Wireless Communications and Mobile Computing

Mobile Edge Computing for 5G and Beyond: Emerging Trends and Applications

Lead Guest Editor: Shaohua Wan
Guest Editors: Sotirios K. Goudos, Maode Ma, Houbing Song, and Shahid Mumtaz

Mobile Edge Computing for 5G and Beyond: Emerging Trends and Applications

Wireless Communications and Mobile Computing

Mobile Edge Computing for 5G and Beyond: Emerging Trends and Applications

Lead Guest Editor: Shaohua Wan
Guest Editors: Sotirios K. Goudos, Maode Ma, Houbing Song, and Shahid Mumtaz

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9858963, 1 page
https://doi.org/10.1155/2023/9858963

Retraction

Retracted: Improved Multiview Decomposition for Single-Image High-Resolution 3D Object Reconstruction

Wireless Communications and Mobile Computing

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Peng, K. Fu, Q. Wei, Y. Qin, and Q. He, "Improved Multiview Decomposition for Single-Image High-Resolution 3D Object Reconstruction," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8871082, 14 pages, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6692390, 11 pages
https://doi.org/10.1155/2021/6692390

Research Article

A Robust Data-Driven Method for Multiseasonality and Heteroscedasticity in Time Series Preprocessing

Bin Sun,[1] **Liyao Ma**,[1] **Tao Shen**,[1] **Renkang Geng**,[1] **Yuan Zhou**,[2] **and Ye Tian**[3]

[1]*School of Electrical Engineering, University of Jinan, Jinan 250022, China*
[2]*Blekinge Institute of Technology, Karlskrona 37179, Sweden*
[3]*China Information Communication Technologies Group Corporation (CICT), Wuhan 430074, China*

Correspondence should be addressed to Liyao Ma; cse_maly@ujn.edu.cn and Tao Shen; cse_st@ujn.edu.cn

Received 27 November 2020; Accepted 28 July 2021; Published 16 August 2021

Academic Editor: Shaohua Wan

Internet of Things (IoT) is emerging, and 5G enables much more data transport from mobile and wireless sources. The data to be transmitted is too much compared to link capacity. Labelling data and transmit only useful part of the collected data or their features is a promising solution for this challenge. Abnormal data are valuable due to the need to train models and to detect anomalies when being compared to already overflowing normal data. Labelling can be done in data sources or edges to balance the load and computing between sources, edges, and centres. However, unsupervised labelling method is still a challenge preventing to implement the above solutions. Two main problems in unsupervised labelling are long-term dynamic multiseasonality and heteroscedasticity. This paper proposes a data-driven method to handle modelling and heteroscedasticity problems. The method contains the following main steps. First, raw data are preprocessed and grouped. Second, main models are built for each group. Third, models are adapted back to the original measured data to get raw residuals. Fourth, raw residuals go through deheteroscedasticity and become normalized residuals. Finally, normalized residuals are used to conduct anomaly detection. The experimental results with real-world data show that our method successfully increases receiver-operating characteristic (AUC) by about 30%.

1. Introduction

Together with rapid development of 5G, the connection requirement of wireless devices is also developing due to the eased connectivity and much shorter (in milliseconds) delay. A result is that Internet of Things (IoT) technologies are now used by more than a quarter of mainstream business compared to 13% six years ago. A great number of industry companies started to put attention on their IoT time series data, including but not limited to health care [1] and transportation [2]. While lots of mobile vehicles are connected to the IoT network as data sources [3], much more data is produced. On one aspect, it is an opportunity for machine learning-based data processing methods. On the other aspect, data transmission is now more challenging.

Moving and remote data source create a challenge that it is hard to send data, especially using wireless ways, as it is still expensive to use limited wireless resource to transfer data even for 5G service providers. In some situations, if real-time moving vehicle information is needed while radio signal is limited, then wireless and wired connection may be both needed to provide support together [4, 5]. This situation is shown in Figure 1.

For this situation, one way to solve it is to label data near to sources. It not only reduces the amount of data to transfer but also balances the computing load between edges and centres [6]. One more benefit is that labelling different types of data is good for later prediction [7]. However, most solutions require labelled data to train labelling models or human expert rich experience to configurate parameters.

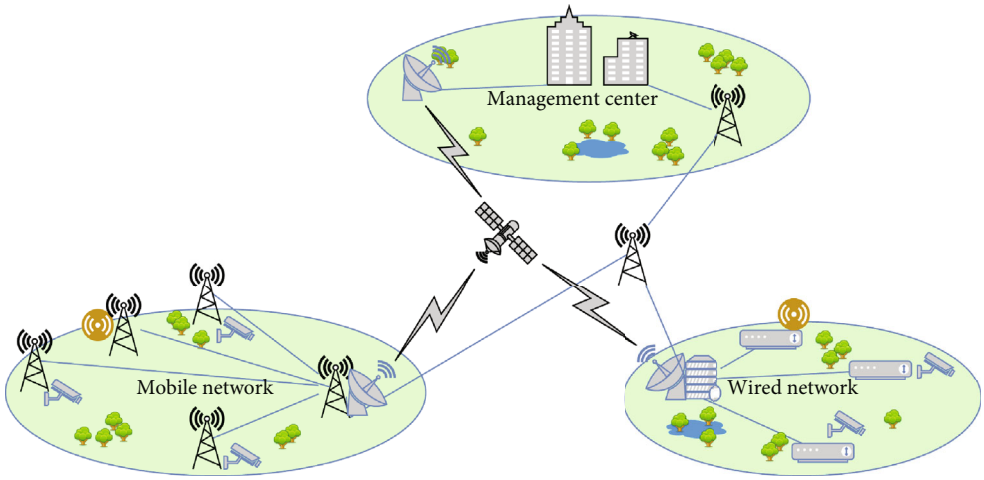


Figure 1: Data transmission across multiple 5G wireless and wired networks among sources, edges, and centres.

In this work, we try to solve this problem by enhancing data preprocessing. Our previous initial feasibility experiments show promising results [8] and we complete the design here. The main contributions of this work include detailed steps of the data-driven method to handle heteroscedasticity of Internet of Things (IoT) data and comparison of possible unsupervised labelling methods as well as analysis of the reasons.

The remaining content is organized as follows. First, the section introduces the problem and related definitions, together with previous research that tried to tackle this problem. Second, the proposed method is thoroughly documented in the section including steps of data preprocessing, model building, model adaptation, residual matrix construction, and anomaly detection. Third, the section describes a series of experiments using real-world data that are carried out in order to evaluate and compare the performance of the proposed method in terms of different metrics. Finally, experimental results are shown and analysed in the section, and conclusions are made in the section.

2. Background and Related Work

Here, we consider a system with a centre node. IoT data processing happens across the entire system [9]. It starts as early as the source application data part as shown in the updated TCP/IP architecture in Figure 2. Example source application data include camera images, video streaming, temperature, and other environmental sensed values [10]. The sensed data are then sent via possible networking routing which could be fully used for distributed processing [11], especially together with the application layer [12–14]. Physical layer choice matters as the emergency level and importance level differ among transported data which should be optimized carefully [15, 16]. When the data finally arrive at the centre, data mining algorithms could be applied [17] to analyse and conduct prediction in most cases.

Regarding labelling and detection of anomalies in time series, much work has been done. Previous work can be categorized in different ways from different aspects [18]. A typical categorization includes the following categories. Probability-based methods calculate a density distribution and use some kind of thresholds to the distribution centre to label anomalies [19]. Distance-based methods set thresholds regarding how far an instance deviates from its neighbours. The measurement can be defined distances, such as in k-nearest neighbours [20], or some kind of cost of separation such as decision tree-based methods [21]. Reconstruction-based methods catch patterns and calculate the expected values of instances to get the difference, i.e., residuals, and then use residuals to conduct labelling [22, 23]. Boundary-based methods, such as support vector machine [24], provide a boundary or hyperplane to separate abnormal instances from normal ones. In addition, ensemble methods can be used to improve the accuracy and robustness of above methods [25]. For the above-mentioned methods, reconstruction-based methods give not only residuals but also comprehensive patterns and models. Thus, this work focuses on providing a preprocessing procedure to calculate and standardize residuals as the first step of reconstruction-based methods.

For reconstructed residuals, as the original saved data is huge and long-term, one common problem is the variance of residuals are time-dependent, i.e., heteroscedasticity [26]. Using traffic flow as an example, the variance is high during noon time when the flow itself is high as shown in Figure 3. Vice versa, the flow and its variance are both low after midnight. This causes problems for labelling algorithms as many of them cannot distinguish high variances with anomalies.

During literature review, we found two methods that try to solve the above two problems at the same time. One method is SARIMA-GARCH (Seasonal Auto-Regressive Integrated Moving Average-Generalized Auto-Regressive Conditional Heteroscedasticity) [27]. Another one is TBATS (Trigonometric Box-cox transform, ARMA errors, Trend and Seasonal component) [26]. Thus, those two methods are also tested in this work. For the final detection part, SHESD (Seasonal Hybrid Extreme Studentized Deviate test) [28] shows promising results in experiments [29–31] and is

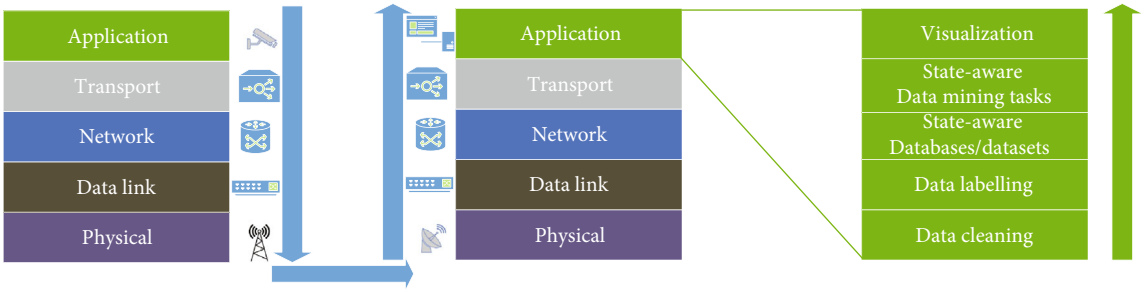


Figure 2: Data flow and process architecture.

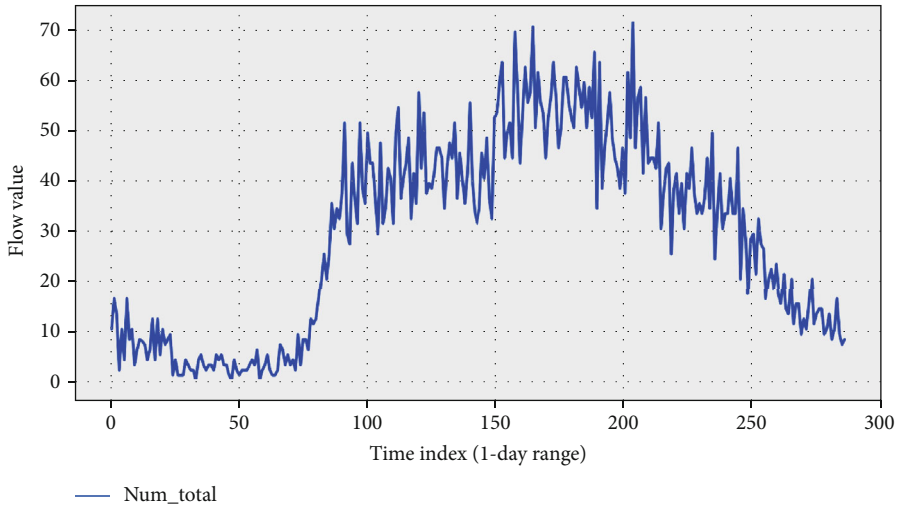


Figure 3: Overview of typical time series with heteroscedasticity.

used here. It is worth mentioning that there are plenty of alternative methods while this work focuses on preprocessing.

3. Methodology

The proposed method includes three main steps which are preprocessing, building day-of-week (DOW) models, and solving flow-level-heteroscedasticity problem. This part describes the method in detail. The entire procedure is summarized in Figure 4.

3.1. Preprocess Data. In this part, data are loaded and then divided into seven groups according to day of week.

For consecutive zeros (continuous three or more zeros) which means controlled access or device malfunction, set flags and replace the instances with *null*:

$$v_i^{\text{flag}} = \begin{cases} 1, & \text{if } r_i \text{ is in consecutive zeros,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where r_i is the ith measured flow rate value.

Instead of using original natural daily periods, we use a new starting point. The purpose is to find a base where the starting flow rates of seasons are low and similar so that the robust fitting could work better in latter steps. It is worth mentioning that (daily) seasons may start from other time than midnight. Actually, the starting point is calculated to be around 3 am in the experiments.

$$N^{\text{seasons}} = \left\lfloor \frac{N^{\text{origin}}}{N^{\text{periods}}} \right\rfloor, \quad (2)$$

where N^{seasons} is the number of complete seasons, N^{origin} is the original number of instances (about 288×406 days), and N^{periods} is the number of periods (i.e., instances) per day (e.g., 288 per day for 5-minute interval data).

All complete seasons are put together to construct a matrix:

$$R = \left[s_1\ s_2\ \cdots\ s_{i_s}\ \cdots\ s_{N^{\text{seasons}}} \right], \quad (3)$$

$$= \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,i_s} & \cdots & r_{1,N^{\text{seasons}}} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,i_s} & \cdots & r_{2,N^{\text{seasons}}} \\ \vdots & \vdots & \ddots & & & \vdots \\ r_{i_p,1} & r_{i_p,2} & & r_{i_p,i_s} & & r_{i_p,N^{\text{seasons}}} \\ \vdots & \vdots & & & \ddots & \vdots \\ r_{N^{\text{periods}},1} & r_{N^{\text{periods}},2} & \cdots & r_{N^{\text{periods}},i_s} & \cdots & r_{N^{\text{periods}},N^{\text{seasons}}} \end{bmatrix}, \quad (4)$$

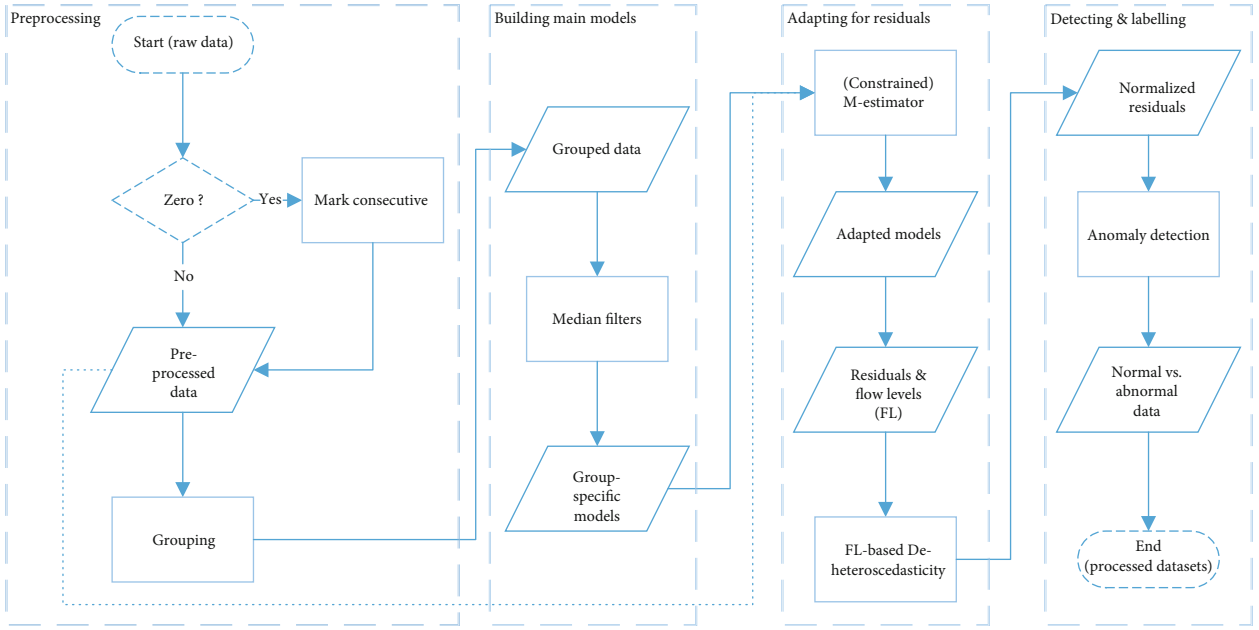


FIGURE 4: Summary of the proposed procedure.

with each season constructing a column, e.g., $s_1 = [r_1, r_2, \cdots, r_{N^{\text{periods}}}]^T$.

Then, separate the seasons/columns into groups; here, we use the day of week of the season starting point as the criteria; thus, there are 7 groups $(G_1, \cdots G_7)$ with similar number of instances for each group.

$$G_i = \{s_{7n+i} \mid n = 0, 1, 2, \cdots \text{and } 7n + i \le N^{\text{seasons}}\}. \quad (5)$$

3.2. Build the Main Models. Now, seven day-of-week (DOW) models are built with the key concept of median. The building algorithm is designed in the way that it can set up several workers in parallel to improve building performance.

To get a specific model $\mathbf{M}_i$, a matrix $\check{\mathbf{R}}_{i_m}$ is constructed by using all seasons (all columns) of G_i:

$$\check{\mathbf{R}}_{i_m} = \begin{bmatrix} \check{r}_{1,1} & \check{r}_{1,2} & \cdots & \check{r}_{1,i_r} & \cdots & \check{r}_{1,N^{M_i}} \\ \check{r}_{2,1} & \check{r}_{2,2} & \cdots & \check{r}_{2,i_r} & \cdots & \check{r}_{2,N^{M_i}} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ \check{r}_{i_p,1} & \check{r}_{i_p,2} & & \check{r}_{i_p,i_r} & & \check{r}_{i_p,N^{M_i}} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \check{r}_{N^{\text{periods}},1} & \check{r}_{N^{\text{periods}},2} & \cdots & \check{r}_{N^{\text{periods}},i_r} & \cdots & \check{r}_{N^{\text{periods}},N^{M_i}} \end{bmatrix}. \quad (6)$$

N^{M_i} is the number of complete seasons for a specific model m_i.

Seven DOW models M_i^{dow} $(i = 1, \cdots, N^{\text{models}}$ where N^{models} is 7 in this paper) are built by applying median filters to $\check{\mathbf{R}}_{i_m}$.

We can present all models as columns of a matrix:

$$M = \begin{bmatrix} m_1 & m_2 & \cdots & m_{i_m} & \cdots & m_{N^{\text{models}}} \end{bmatrix}, \quad (7)$$

$$= \begin{bmatrix} \tilde{r}_{1,1} & \tilde{r}_{1,2} & \cdots & \tilde{r}_{1,i_m} & \cdots & \tilde{r}_{1,N^{\text{models}}} \\ \tilde{r}_{2,1} & \tilde{r}_{2,2} & \cdots & \tilde{r}_{2,i_m} & \cdots & \tilde{r}_{2,N^{\text{models}}} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ \tilde{r}_{i_p,1} & \tilde{r}_{i_p,2} & & \tilde{r}_{i_p,i_m} & & \tilde{r}_{i_p,N^{\text{models}}} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \tilde{r}_{N^{\text{periods}},1} & \tilde{r}_{N^{\text{periods}},2} & \cdots & \tilde{r}_{N^{\text{periods}},i_m} & \cdots & \tilde{r}_{N^{\text{periods}},N^{\text{models}}} \end{bmatrix}, \quad (8)$$

where the $i_m = 1, 2, \cdots, N^{\text{models}}$ indicates model index and the $i_p = 1, 2, \cdots, N^{\text{periods}}$ indicates time point (period) index of day. Thus,

$$\tilde{r}_{i_p,i_m} = \text{med}\left(\text{row}_{i_p} \check{R}_{i_m}\right), \quad (9)$$

where $\check{R}_{i_p,i_m} = \{\check{r}_i \in R\}$, i.e., $\check{R}_{i_p,i_m} \subset R$ and contains all r's with the time point index of day i_p which belongs to model M_{i_m}.

3.3. Adapt like Regressors. This part calculates fitted models using M-estimation considering the above model matrix and each individual season.

An M-estimator is then computed iteratively with reweighted least squares (IRLS):

$$\beta^{(t+1)} = \arg\min_{\beta} \sum_{i_p=1}^{N^{\text{periods}}} w_{i_p}\left(\beta^{(t)}\right)\left|\varepsilon_{i_p}(\beta)\right|^2, \quad (10)$$

where the scaling and addition parameters $\beta = [k, b]$, and residuals from the previous fit (using season i_s belongs to model i_m as an example):

$$\varepsilon(\beta) = s_{i_s} - f_a\left(m_{i_m}, \beta_{i_s,i_m}\right) = col_{i_s} R - \left(k_{i_s,i_m}\, col_{i_m} M + b_{i_s,i_m}\right). \tag{11}$$

Thus, the residual matrix:

$$E = \begin{bmatrix} \varepsilon_{1,1} & \varepsilon_{1,2} & \cdots & \varepsilon_{1,i_s} & \cdots & \varepsilon_{1,N^{\text{seasons}}} \\ \varepsilon_{2,1} & \varepsilon_{2,2} & \cdots & \varepsilon_{2,i_s} & \cdots & \varepsilon_{2,N^{\text{seasons}}} \\ \vdots & \vdots & \ddots & & & \vdots \\ \varepsilon_{i_p,1} & \varepsilon_{i_p,2} & & \varepsilon_{i_p,i_s} & & \varepsilon_{i_p,N^{\text{seasons}}} \\ \vdots & \vdots & & & \ddots & \vdots \\ \varepsilon_{N^{\text{periods}},1} & \varepsilon_{N^{\text{periods}},2} & \cdots & \varepsilon_{N^{\text{periods}},i_s} & \cdots & \varepsilon_{N^{\text{periods}},N^{\text{seasons}}} \end{bmatrix}. \tag{12}$$

During the estimation, the weights are calculated as:

$$w = \left[w_1, \cdots, w_{i_p}, \cdots, w_{N^{\text{periods}}}\right] = \frac{\psi_\gamma(\varepsilon/c)}{\varepsilon}, \tag{13}$$

where c is a scaling factor:

$$c = \frac{\text{med}(abs(\varepsilon))}{\eta}, \tag{14}$$

and ψ is in Huber family:

$$\psi_\gamma(x) = \begin{cases} x, & \text{if } |x| \le \gamma, \\ \gamma \ \text{sign}\ (x), & \text{if } |x| > \gamma, \end{cases} \tag{15}$$

while η is a constant 0.675 and γ is 1.345 which correspond to regression estimator 95% efficiency. If M-estimation fails (rarely), then constrained M-estimation (CM) [32] is used (which is always working for our data). CM is proposed by Mendes and Tyler for regression and is more robust while keeping the same breakdown point (i.e., 1/2) though slower.

3.4. Construct the Residual Matrix. While having the adapted models, the raw residuals can be calculated directly. However, the raw residuals contain different variations on different flow levels. Thus, this part also removes flow-level-related heteroscedasticity.

For adapted models, i.e., $f_a(m, \beta)$, let us take values of adapted models and round them to integers then we get flow levels as integers l of each time point.

$$L = \lfloor f_a(m, \beta) \rceil, \tag{16}$$

$$= \left[\hat{s}_1 \quad \hat{s}_2 \quad \cdots \quad \hat{s}_{i_s} \quad \cdots \quad \hat{s}_{N^{\text{seasons}}}\right], \tag{17}$$

$$= \begin{bmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,i_s} & \cdots & l_{1,N^{\text{seasons}}} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,i_s} & \cdots & l_{2,N^{\text{seasons}}} \\ \vdots & \vdots & \ddots & & & \vdots \\ l_{i_p,1} & l_{i_p,2} & & l_{i_p,i_s} & & l_{i_p,N^{\text{seasons}}} \\ \vdots & \vdots & & & \ddots & \vdots \\ l_{N^{\text{periods}},1} & l_{N^{\text{periods}},2} & \cdots & l_{N^{\text{periods}},i_s} & \cdots & l_{N^{\text{periods}},N^{\text{seasons}}} \end{bmatrix}. \tag{18}$$

Be aware that the flow levels are rounded from adapted model values instead of measured. For example, suppose 9 am traffic is 85 in the DOW model, 90.3 in the adapted model, but only 10 in the measured traffic (due to an incident or so); then, the traffic flow level is 90, i.e., flow level is a adapted and generalized description which represents what the traffic should be during a similar day.

Suppose the minimum and maximum integers (levels) in L are:

$$l_{\min} = \min\left(l_{i_p,i_s}\right), l_{\max} = \max\left(l_{i_p,i_s}\right), \quad l_{i_p,i_s} \in L, \tag{19}$$

then we can generate a level vector

$$\breve{L} = \left[\, l_{\min} \quad l_{\min} + 1 \quad \cdots \quad l_{i_l} \quad \cdots \quad l_{\max} \,\right], \tag{20}$$

$$= \left[\, \breve{l}_1 \quad \breve{l}_2 \quad \cdots \quad \breve{l}_{i_l} \quad \cdots \quad \breve{l}_{N^{\text{levels}}} \,\right], \tag{21}$$

which contains all integers from $l_{\min}$ to $l_{\max}$ and $N^{\text{levels}} = l_{\max} - l_{\min} + 1$ denotes the number of total flow levels.

For all level items/values in adapted models L, adapted models do element-wise XNOR logic and we get a mask matrix A with ones indicating the time points/instances with flow levels of l_{i_l}.

$$A_{i_l} = l_{i_l} \bar{\oplus} L = \left\{ a_{i_a,j_a} \mid i_a = 1, 2, \cdots, N^{\text{periods}}; j_a = 1, 2, \cdots, N^{\text{seasons}} \right\}, \tag{22}$$

$$a_{i_a,j_a} = \begin{cases} 1, & \text{if } l_{i_p,i_s} = l_{i_l}, \\ \text{null}, & \text{otherwise.} \end{cases} \tag{23}$$

Let us apply this mask A_{i_l} to E and take all the matched values then calculate the variance (standard deviation) for an arbirtary level *null* items and related calculation are ignored during this process.

$$\breve{E}_{i_l} = A \cdot E = \left\{ \varepsilon_{i_p,i_s} \mid a_{i_p,i_s} = 1 \right\}, \tag{24}$$

$$v_{i_l} = std\left(\breve{E}_{i_l}\right). \tag{25}$$

The variances for different levels vary, thus heteroscedasticity. When putting all variances for all levels to get a variance/-heteroscedasticity vector, note that residuals from neighbour levels are used when the amount of residuals is insufficient.

```
1: procedure DOW-FLH (Original Time Series)
2:     set flags for consecutive zeros ▹Handel Dirty Data
3:     for each day do
4:         find the time point index (TPI) of the lowest flow
5:     end for
6:     find TPIs' median number as starts of daily seasons, e.g., 3 am
7:     for model m_{i_m} in all DOW models do▹Build DOW Models
8:         take all seasons related to m_{i_m} to a group
9:         remove flagged consecutive zeros
10:        calculate median of grouped seasons as the model m_{i_m}
11:    end for
12:    for model m_{i_m} in all DOW models do ▹Fit/Adapt to Get Scalings k and Additions b
13:        for each realted season do
14:            remove flagged consecutive zeros
15:            estimate k, b by robustly fitting m_{i_m} to the season
16:            rounding all values of the fitted model to integers as the season's flow levels
17:            get residuals as the difference between the fitted and the season
18:        end for
19:    end for
20:    for each flow level Standardize Residuals (FLH) do▹Standardize Residuals (FLH)
21:        take all residuals for this flow level (or with neighbours if not enough)
22:        calculate standard deviations (STD)
23:    end for
24:    consider all STDs with all flow levels as the flow level heteroscedasticity (FLH)
25:    divide each residual with timely corresponding STD to standardize
26:    for each detection algorithm do ▹Detection
27:        feed the entire standardized residual time series to the algorithm
28:        get algorithm-specific anomalies or anomaly scores
29:    end for
30:    return the list of anomalies or anomaly scores
31: end procedure
```

Algorithm 1: DOW-FLH Modelling for Data Preprocessing

$$V = \begin{bmatrix} v_1 & v_2 & \cdots & v_{i_l} & \cdots & v_{N^{\text{levels}}} \end{bmatrix}. \tag{26}$$

Later, all residuals **E** are divided by the time point's level's variance to get "normalized residuals." First, for levels of each time point, i.e., l_{i_p,i_s}, find its corresponding variance:

$$\hat{i}_l(i_p, i_s) = \arg \underset{i_l}{\text{where}}\, l_{i_p,i_s} = \breve{l}_{i_l}, \tag{27}$$

$$\hat{v}_{i_p,i_s} = v_{\hat{i}_l(i_p,i_s)}. \tag{28}$$

Generate a matrix of all residual's corresponding variance:

$$\hat{V} = \begin{bmatrix} \hat{v}_{1,1} & \hat{v}_{1,2} & \cdots & \hat{v}_{1,i_s} & \cdots & \hat{v}_{1,N^{\text{seasons}}} \\ \hat{v}_{2,1} & \hat{v}_{2,2} & \cdots & \hat{v}_{2,i_s} & \cdots & \hat{v}_{2,N^{\text{seasons}}} \\ \vdots & \vdots & \ddots & & & \vdots \\ \hat{v}_{i_p,1} & \hat{v}_{i_p,2} & & \hat{v}_{i_p,i_s} & & \hat{v}_{i_p,N^{\text{seasons}}} \\ \vdots & \vdots & & & \ddots & \vdots \\ \hat{v}_{N^{\text{periods}},1} & \hat{v}_{N^{\text{periods}},2} & \cdots & \hat{v}_{N^{\text{periods}},i_s} & \cdots & \hat{v}_{N^{\text{periods}},N^{\text{seasons}}} \end{bmatrix}. \tag{29}$$

Normalized residuals are:

$$R' = \left\{ r'_{i_p,i_s} \mid i_p = 1, 2, \cdots, N^{\text{periods}}; i_s = 1, 2, \cdots, N^{\text{seasons}} \right\}, \tag{30}$$

where

$$r'_{i_p,i_s} = \frac{r_{i_p,i_s}}{\hat{v}_{i_p,i_s}}. \tag{31}$$

3.5. Detect Using Normalized Residuals. Finally, normalized residuals are sent to detection algorithms. The entire procedure is also presented in pseudocode (Algorithm 1).

4. Experiments

This section describes data, practical procedure, and the way we conduct experiments.

4.1. Data Specification. The one-year long real-world data are collected from a highway. Ground truth anomaly (incidents) labels are generated by using the extended system mentioned in [33]. The data are imputed using the method from [34] before any processing. One device sends a monitored flow

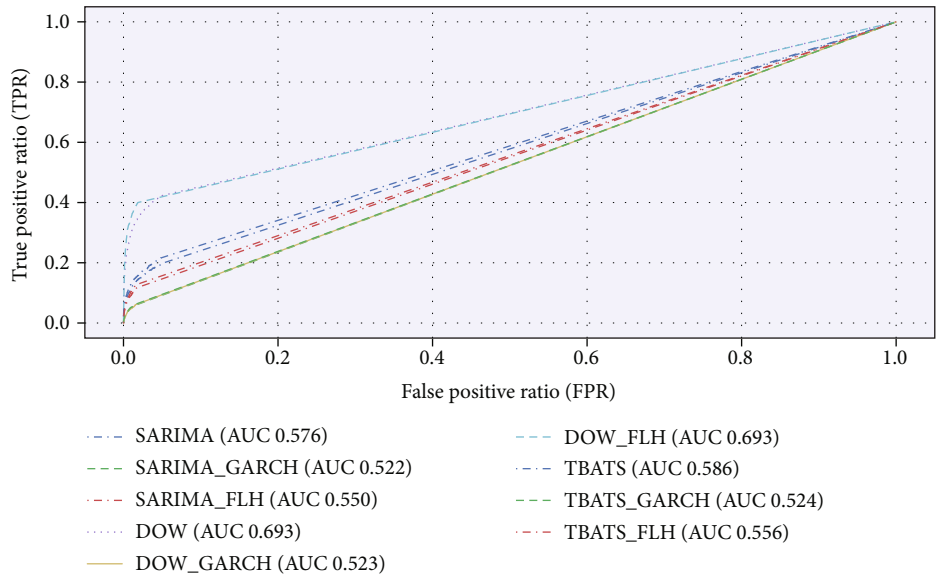


Figure 5: The proposed DOW and DOW-FLH gives similar AUC and around 27% bigger coverage than the others on average. Besides, DOW-FLH produces only half false positives compared to DOW without FLH.

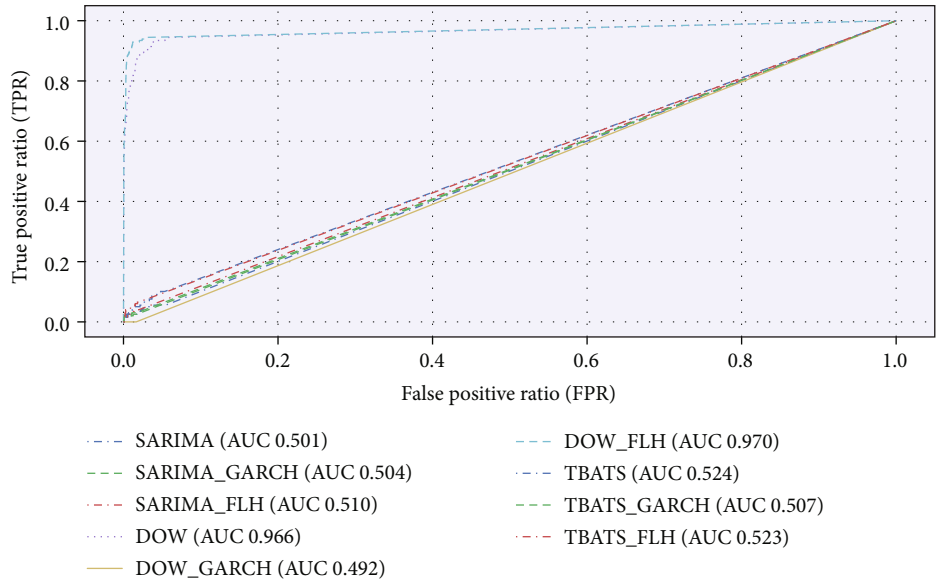


Figure 6: Only DOW-based algorithms can detect device malfunction efficiently while others are close to no discrimination line.

record at five-minute intervals. Each record contains some traffic statistics such as flow rate and average speed. This road carries undersaturated flow except in holidays' noons, where is 15 min average?

4.2. Experimental Setup. The experiments are done in a desktop computer with AMD Ryzen 5-3600 (6 Cores, 3600 MHz) and 16 GB DDR4 memory. To be fair, we only implement our method; other algorithms are taken from public domain such as GitHub.

Our implementation is done in the R programming environment version 3.4.3 with RStudio 1.3.1056, AnomalyDetection 1.0, forecast 8.2, feather 0.3.3 as well as the Python programming environment version 3.6.7/3.6.9 with library arch 4.8.1, statsmodels 0.9.0, feather-format 0.4.0/0.4.1, numpy 1.16.0/1.19.4, pandas 0.23.4/1.1.4, scikit-learn 0.19.2/0.23.2, scipy 1.2.2/1.5.4, ipykernel 5.3.4, and ipython 7.16.1.

SHESD was originally implemented to give only binary results so we modified it by adding $\text{test}_{r}\text{esult} - \text{critical}_{v}\text{alue}$ to get anomaly/outlier scores. Also, as the max allowed anomaly (outliers) ratio is 50%, we mark all nontested ones the same score as the lowest score.

4.3. Evaluation Measurement and Metrics. Receiver-operating characteristic (ROC) is used as the main evaluation metric as it provides an accurate and visualized way to present detecting results. One important value from ROC is area

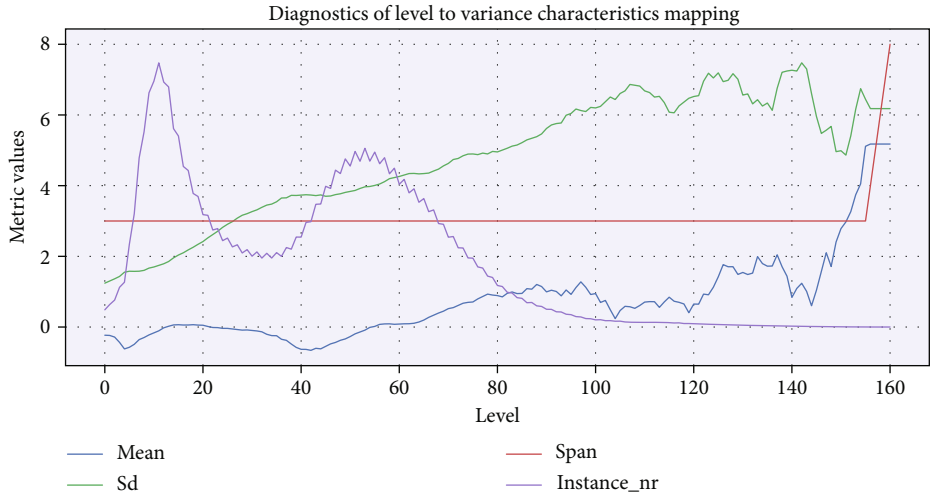


Figure 7: Flow level with corresponding instance numbers and mean, STD, and span of residuals (instance number per level is scaled down to be shown in this figure).

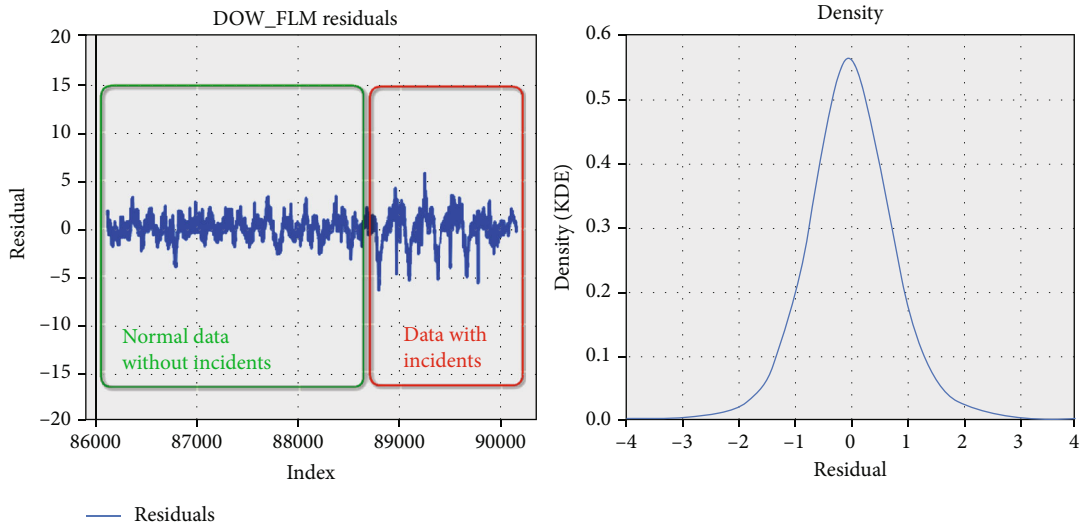


Figure 8: The left one shows DOW-FLH residuals of time series data without and with abnormal data and the right one shows overall residual density for the entire dataset. DOW-FLH successfully suppressed heteroscedasticity for normal data.

under curve (AUC) which is also known as A′ ("a-prime"), or concordance-statistic (*c*-statistic). It is a measure of goodness of fit that is often used for binary classification modelling results evaluation; therefore, we use it here.

5. Results and Analysis

As shown in Figure 5, our DOW and DOW-FLH methods are superior with regard to AUC. DOW with and without FLH performs similar considering AUC of 0.693 from both algorithms which are 26.9% better coverage than other algorithms on average (AUC 0.546). What is more, DOW-FLH is preferred for less false positives on the optimal cut-off point compared to DOW without FLH due to the data sensitivity to false positive. May move below to analysis? For unbalanced datasets such as traffic flows, this behaviour gives positive influence. The reason is that some false-negative instances introduce only minor issues for true-negative ones as negative instances are majority while the same amount false-positive instances impact true anomalies (incidents) much more.

We analysed the detection ratio and AUCs for different situations and found some interesting results. For device malfunction incidents, most algorithms cannot notice it as shown in Figure 6. The possible reason is that other algorithms are tracking no-flow situation without considering normal situation. Note that good seasonal modelling (DOW) should work with suitable variance handling methods, as inappropriate variation handling (i.e., GARCH) may otherwise reduce the effectiveness.

Figure 7 shows level to residual characteristics diagnostics. The mean of residuals (blue line) is mostly under 2 but

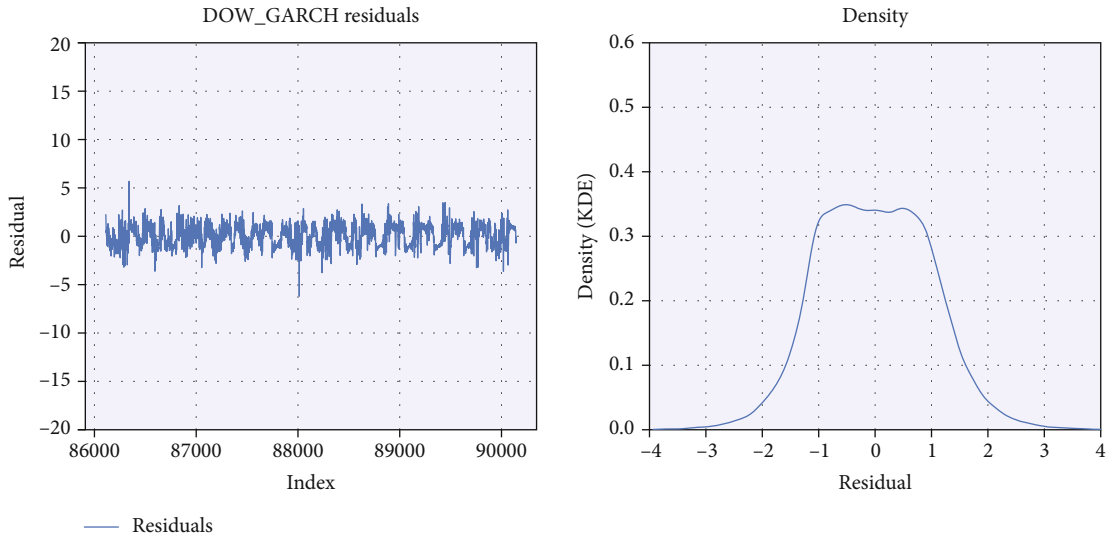


Figure 9: DOW-GARCH suppresses not only extreme values and heteroscedasticity for data with all normal instances but also for data with abnormal instances, when being compared to DOW-FLH (Figure 9) (same data range).

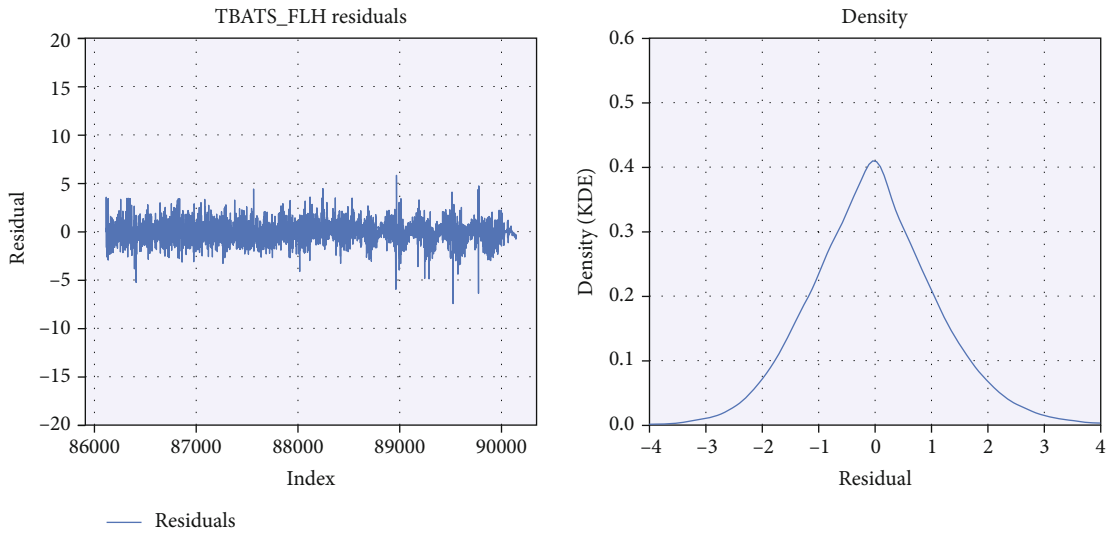


Figure 10: TBATS-based algorithms produce residuals with much noise when being compared to DOW-FLH (Figure 9) (same x-axis time range). (S)ARIMA-based algorithms give similar results.

increases rapidly to be about 5 when the flow level is greater than 150. This is due to the fact that extreme levels (greater than 150) occur only during few big holidays, so this scenario is hard to be caught by models. The standard deviations (green line) is mainly increasing which represents one key problem, i.e., heteroscedasticity. The purple line represents the number of instances per level, and it becomes very small for extreme scenarios in both directions of x-axis. The number of span is used to include neighbour levels when one level's corresponding instances are too few to calculate reasonable statistics. In summary, it can be seen that the relation mapping from levels to residual characteristics are nonlinear. This explains why the proposed data-driven algorithms perform better.

DOW successfully modelled patterns and FLH successfully suppressed heteroscedasticity for normal data compared to others as the residuals are shown in Figure 8. Other algorithms, when being compared to DOW-FLH, cannot distinguish data with vs. without abnormalities, such as shown in Figure 9. This could be an advantage for GARCH-based methods when tracing rapid change in (nonseasonal) time series with heteroscedasticity, but it becomes an disadvantage and hides possible abnormal data instances here. The problem with TBATS and SARIMA is that they could not successfully model the patterns and produces residuals with much noise which leads to low signal-noise ratio as shown in Figure 10.

Previous work has shown that ARIMA and GARCH cannot be adapted to seasonality with many periods such as here

288 periods per season. Instead, they will adapt to local trend or rapid change add plots; therefore, they are not suitable to detect anomalies lasting beyond their detection abilities. This characteristic could be an advantage when quick predicting traffic for short-term time is needed.

6. Conclusion and Future Work

The experiment results show that the proposed DOW algorithm is good at matching multiseasonality time series patterns, and FLH can solve heteroscedasticity problem. DOW-FLH-modelled residuals can be used for labelling anomalies; then, the chosen data can be sent to either edges or centres for further process.

As discussed above, the proposed DOW-FLH in this work is good at modelling and labelling multiseasonal IoT time series for the edge-centre structure. However, other compared algorithms, including SARIMA- and TBATS-based ones, are more mature and may be good at local trend prediction. Also, edge computing can engage crowdsourcing and related active learning [35] to make full use of advantages provided by edge-centre structure.

This point can be further tested in later research.

Labelling can be treated as a classification question, and many new algorithms can work on this task. Especially, recent development regarding classification using belief theory is showing promising results [36], and it is good for multisource scenarios in edge-centre computing. Thus, this might be a good enhancement for our current work, and we look forward to investigate more about it in the future work.

In summary, the proposed DOW-FLH method performs well during experiments using multiseasonal IoT time series and should be considered to use when labelling is needed in edge-centre computing structure.

Data Availability

Access to data is restricted in general; please contact the authors for access when necessary.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We would like to thank the reviewers for their detailed suggestions that have indeed made great contribution to this work. This work is supported by the Shandong Natural Science Foundation under ZR2020MF067 and Shandong Key Research and Development Program under 2019JZZY021005.

References

[1] M. Sun, C. Smith, D. Howard et al., "FES-UPP: a flexible functional electrical stimulation system to support upper limb functional activity practice," *Frontiers in Neuroscience*, vol. 12, 2018.

[2] B. Mandler, J. Barja, M. E. M. Campista et al., *Internet of Things-IoT Infrastructures, Volume 169*, Springer, 2016.

[3] Q. Liu, K. M. Kamoto, X. Liu et al., "A sensory similarities approach to load disaggregation of charging stations in Internet of electric vehicles," *IEEE Sensors Journal*, vol. 9, 2020.

[4] Y. Xu, C. K. Ahn, Y. S. Shmaliy, X. Chen, and Y. Li, "Adaptive robust INS/UWB-integrated human tracking using UFIR filter bank," *Measurement*, vol. 123, pp. 1–7, 2018.

[5] Y. Xu, Y. S. Shmaliy, C. K. Ahn, T. Shen, and Y. Zhuang, "Tightly-coupled integration of INS and UWB using fixed-lag extended UFIR smoothing for quadrotor localization," *IEEE Internet of Things Journal*, vol. 8, no. 3, p. 1, 2020.

[6] M. Sun, X. Zhu, P. Zhang et al., "Xxx an EEG signal-based music treatment system for autistic children," *Security and Communication Networks*, vol. 2020, Article ID 8868311, 2021.

[7] B. Sun, W. Cheng, P. Goswami, and G. Bai, "Flow-aware WPT k-nearest neighbours regression for short-term traffic prediction," in *22nd IEEE symposium on computers and communication (ISCC)*, pp. 48–53, Heraklion, Greece, July 2017.

[8] B. Sun, W. Cheng, L. Ma, and G. Prashant, "Anomaly-aware traffic prediction based on automated conditional information fusion," in *International conference on information FUSION (FUSION)*, pp. 2283–2289, Cambridge, United Kingdom, July 2018.

[9] K. Jia, Z. Wang, S. Fan, S. Zhai, and G. He, "Data-centric approach: a novel systematic approach for cyber physical system heterogeneity in smart grid," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 14, no. 5, pp. 748–759, 2019.

[10] P. N. Borza, M. Machedon-Pisu, and F. Hamza-Lup, "Design of wireless sensors for IoT with energy storage and communication channel heterogeneity," *Sensors*, vol. 19, no. 15, article 3364, 2019.

[11] C. Chen, Y. Zhang, and W. Zheng, "Distributed computation offloading method based on deep reinforcement learning in ICV," *Applied Soft Computing*, vol. 103, article 107108, 2021.

[12] M. Diyan, B. N. Silva, J. Han, Z. B. Cao, and K. Han, "Intelligent Internet of Things gateway supporting heterogeneous energy data management and processing," *Transactions on Emerging Telecommunications Technologies*, vol. 3, article 3919, 2020.

[13] K. Sood, K. K. Karmakar, S. Yu, V. Varadharajan, S. R. Pokhrel, and Y. Xiang, "Alleviating heterogeneity in SDN-IoT networks to maintain QoS and enhance security," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5964–5975, 2020.

[14] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the heterogeneity: a self-organized federated learning framework for IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3088–3098, 2021.

[15] S. K. Goudos, P. Sarigiannidis, P. I. Dallas, and S. Kyriazakos, "Communication protocols for the IoT-based smart grid," in *IoT for Smart Grids: Design Challenges and Paradigms, Power Systems*, K. Siozios, D. Anagnostos, D. Soudris, and E. Kosmatopoulos, Eds., pp. 55–83, Springer International Publishing, Cham, 2019.

[16] C. Chen, J. Li, V. Balasubramaniam, Y. Wu, Y. Zhang, and S. Wan, "Contention resolution in Wi-Fi 6-enabled Internet of Things based on deep learning," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5309–5320, 2021.

[17] L. Chhaya, P. Sharma, A. Kumar, and G. Bhagwatikar, "Application of data mining in smart grid technology," in

Encyclopedia of Information Science and Technology, pp. 815–827, IGI Global, 2021.

[18] B. Sun and L. Ma, "An overview of outliers and detection methods in general for time series from IoT devices," in *The 10th international conference on computer engineering and networks*, vol. 135, pp. 1180–1186, Xi-An, China, October 2020.

[19] Z. Zheng, H.-Y. Jeong, T. Huang, and J. Shu, "KDE based outlier detection on distributed data streams in multimedia network," *Multimedia Tools and Applications*, vol. 76, no. 17, pp. 18027–18045, 2017.

[20] S.-E. Benkabou, K. Benabdeslem, and B. Canitia, "Unsupervised outlier detection for time series by entropy and dynamic time warping," *Knowledge and Information Systems*, vol. 54, no. 2, pp. 463–486, 2018.

[21] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, p. 3, 2012.

[22] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[23] B. Sun, W. Cheng, P. Goswami, and G. Bai, "Short-term traffic forecasting using self-adjusting k-nearest neighbours," *IET Intelligent Transport Systems*, vol. 12, no. 1, pp. 41–48, 2018.

[24] M. Sun, J. Amor, C. J. James et al., "Methods to characterize the real-world use of rollators using inertial sensors–a feasibility study," *IEEE Access*, vol. 7, pp. 71387–71397, 2019.

[25] L. Ma, B. Sun, and C. Han, "Learning decision forest from evidential data: the random training set sampling approach," in *4th International Conference on Systems and Informatics (ICSAI)*, Hangzhou, China, November 2017.

[26] H. Shi, K. Worden, and E. J. Cross, "A cointegration approach for heteroscedastic data based on a time series decomposition: an application to structural health monitoring," *Mechanical Systems and Signal Processing*, vol. 120, pp. 16–31, 2019.

[27] T. Andrysiak, L. Saganowski, M. Maszewski, and A. Marchewka, "Detection of network attacks using hybrid ARIMA-GARCH model," in *Advances in Dependability Engineering of Complex Systems*, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, Eds., vol. 582, pp. 1–12, Springer International Publishing Ag, Cham, 2018.

[28] J. Hochenbaum, O. S. Vallis, and A. Kejariwal, "Automatic anomaly detection in the cloud via statistical learning," 2017, https://arxiv.org/abs/1704.07706.

[29] S. Kelly and K. Ahmad, "Propagating disaster warnings on social and digital media," in *Intelligent Data Engineering and Automated Learning–IDEAL 2015*, pp. 475–484, Springer, Cham, 2015.

[30] L. Bodrog, M. Kajo, S. Kocsis, and B. Schultz, "A robust algorithm for anomaly detection in mobile networks," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, Valencia, Spain, September 2016.

[31] K. Jensen, T. V. Do, H. T. Nguyen, and A. Arnes, "Better protection of SS7 networks with machine learning," in *2016 6th International Conference on IT Convergence and Security (ICITCS)*, pp. 1–7, Prague, Czech Republic, September 2016.

[32] B. Mendes and D. E. Tyler, "Constrained M-estimation for regression," in *Robust Statistics, Data Analysis, and Computer Intensive Methods: In Honor of Peter Huber's 60th Birthday, Lecture Notes in Statistics*, H. Rieder, Ed., pp. 299–320, Springer New York, New York, NY, 1996.

[33] B. Sun, W. Cheng, G. Bai, and P. Goswami, "Correcting and complementing freeway traffic accident data using mahalanobis distance based outlier detection," *Tehnicki vjesnik - Technical Gazette*, vol. 24, no. 5, pp. 1597–1607, 2017.

[34] B. Sun, L. Ma, W. Cheng, W. Wen, P. Goswami, and G. Bai, "An improved k-nearest neighbours method for traffic time series imputation," in *Chinese automation congress (CAC)*, Jinan, China, October 2017.

[35] L. Ma, S. Destercke, and Y. Wang, "Online active learning of decision trees with evidential data," *Pattern Recognition*, vol. 52, Supplement C, pp. 33–45, 2016.

[36] X. Ke, L. Ma, and Y. Wang, "A dissimilarity measure based on singular value and its application in incremental discounting," in *Proceedings of the 16th International Conference on Information Fusion*, pp. 1391–1397, Istanbul, Turkey, July 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6630492, 12 pages
https://doi.org/10.1155/2021/6630492

Research Article

A Control and Posture Recognition Strategy for Upper-Limb Rehabilitation of Stroke Patients

Xian Yu,[1] **Bo Xiao**,[2] **Ye Tian**,[3] **Zihao Wu**,[4] **Qi Liu**,[4] **Jun Wang**,[4] **Mingxu Sun**,[5] **and Xiaodong Liu**[6]

[1]*NARI Group Co., Ltd. (State Grid Electric Power Research Institute Co., Ltd.), Nanjing 211000, China*
[2]*School of Chemistry and Materials, Nanjing University of Information Science and Technology, Nanjing, Jiangsu 210044, China*
[3]*China Information Communication Technologies Group Corporation (CICT), Wuhan, China*
[4]*School of Computer and Software, Nanjing University of Information Science and Technology, 210044 Nanjing, China*
[5]*School of Electrical Engineering, University of Jinan, China*
[6]*School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh EH10 5DT, UK*

Correspondence should be addressed to Mingxu Sun; cse_sunmx@ujn.edu.cn

Received 18 November 2020; Revised 8 February 2021; Accepted 30 March 2021; Published 18 May 2021

Academic Editor: Jun Cai

At present, the study of upper-limb posture recognition is still in the primary stage; due to the diversity of the objective environment and the complexity of the human body posture, the upper-limb posture has no public dataset. In this paper, an upper extremity data acquisition system is designed, with a three-channel data acquisition mode, collect acceleration signal, and gyroscope signal as sample data. The datasets were preprocessed with deweighting, interpolation, and feature extraction. With the goal of recognizing human posture, experiments with KNN, logistic regression, and random gradient descent algorithms were conducted. In order to verify the superiority of each algorithm, the data window was adjusted to compare the recognition speed, computation time, and accuracy of each classifier. For the problem of improving the accuracy of human posture recognition, a neural network model based on full connectivity is developed. In addition, this paper proposes a finite state machine- (FSM-) based FES control model for controlling the upper limb to perform a range of functional tasks. In the process of constructing the network model, the effects of different hidden layers, activation functions, and optimizers on the recognition rate were experimental for the comparative analysis; the softplus activation function with better recognition performance and the adagrad optimizer are selected. Finally, by comparing the comprehensive recognition accuracy and time efficiency with other classification models, the fully connected neural network is verified in the human posture superiority in identification.

1. Introduction

There are more than 10 million new strokes per year worldwide [1], and stroke is still the leading cause of death and disability among adults [2]. With the accelerating aging of the society and the prevalence of unhealthy lifestyles, stroke diseases have shown explosive growth and are getting younger. Strokes are characterized by high incidence and disability, with World Health Organization data showing that strokes have a disability rate of up to 80%. The economic burden is 10 times greater than that of myocardial infarction. Therefore, prevention and treatment are urgent, and the rehabilitation system for patients needs to be improved.

Stroke patients' recovery of limb function is one of the most important aspects of rehabilitation. At present, there are several different types of rehabilitation therapy in clinic, such as electromyographic feedback therapy, electrical stimulation therapy, and motor imagery mental training therapy, while the most highly regarded in clinical practice is functional electrical functional electrical stimulation (FES), with stimulation electrodes worn on the limbs of stroke patients consisting of the controller send out stimulation signals to electrically stimulate specific muscles to enable the limb to perform various types of functional rehabilitation or to perform daily activity, which in turn leads to the recovery of limb function. Stroke patients need to perform specific

functional tasks in the process of rehabilitation, so an efficient control strategy needs to be designed. At the same time, due to the lack of existing public datasets, it is urgent to establish a database, design algorithms to analyze sensor device data, and identify the upper-limb posture movement of stroke patients. This can provide reference for the rehabilitation and rehabilitation effect of stroke patients. It provides an effective solution.

The paper is divided as follows: Section 2 presents the related work on this field. Section 3 and Section 4 demonstrate the methodologies. Section 5 shows the results and discusses the findings. Finally, Section 6 concludes the paper.

2. Related Work

The related work in this paper concerns FES control and upper-limb posture recognition, and the following sections will focus on these two components.

2.1. Related Work on FES Control. Functional electrical stimulation (FES) is often used for rehabilitation treatment of stroke or spinal cord injury. For individuals with motor nervous system damage, FES can activate the skeletal muscle of paralyzed patients by implementing low-level electrical pulses on motor neurons [3] and activate corresponding muscles according to different expected movements [4]. Since Liberson et al. first used FES to rehabilitate a prolapsed foot in 1960 [5], FES has been proved to be one of the important methods to treat stroke rehabilitation or spinal cord injury. Sabut et al. proposed a combination of FES and general rehabilitation program, which has a significant effect on improving the muscle strength of patients [6]. It is not easy to use FES to control the target skeletal muscle at a high level. When FES stimulates the muscle, the muscle response to the stimulation is nonlinear and time-varying, and individuals with nervous system damage are often accompanied with time delay [7]. There are open-loop, closed-loop, and state machine-based control strategies in FES system to deal with the above problems.

Open-loop control strategy is a simple but reliable control strategy, which is widely used in various control systems. However, due to the low precision of open-loop control and the lack of automatic correction ability, closed-loop control solves this problem [8]. The closed-loop FES control system usually consists of feedback signals, error detection and correction processes, and a model used to determine the output of the system. For example, Zhang et al. proposed an electromyography-based closed-loop torque control strategy of functional electrical stimulation [9], and FES-evoked electromyography (EMG) was used to reflect the state of the stimulated muscle, so as to compensate the muscle strength adaptively. Compared with open-loop control, the closed-loop FES system using surface electronics (sEMG) biases feedback from bilateral arms for enhancing upper-limb stroke rehabilitation [10]. Dodson et al. used a closed-loop controller to compensate for electromechanical delay (EMD) to increase the energy expenditure of the hybrid neural prosthesis and prolong the onset of muscle fatigue [11]. Compared to open-loop control, closed-loop control strategies have better automaticity and adaptivity.

Generally, a FSM controller is composed of a set of states, state transition conditions, input signals, and output functions [12]. Each "state" corresponds to a movement stage, and the "state transition condition" realizes the exit of FES from each movement. Condition, finite state control usually contains multiple states, each action corresponding to each state is predefined, and the transition between states is determined by the current state and artificial signals. Finite state machine has been proved to be an effective control method to realize the functional tasks of upper limbs. For example, the upper-limb auxiliary system designed by Wang et al., which combines FES with robotic exoskeleton, realizes the control of finite state machine based on embedded environment. The finite state machine is designed as an advanced controller, which sends commands to the embedded controller in real time to assist the grasping task realized by the assistant [13]. The experimental results proved the effectiveness of this method.

2.2. Related Work on Posture Recognition. The human body posture recognition mode is divided into vision-based human body posture recognition and sensor-based human body posture recognition [14, 15]. The first one mainly uses support vector machine [16, 17], hidden Markov, and other algorithms [18]. The recognition success rate or the efficiency of the algorithm is ideal, but it is more environment dependent, the conditions are limited, and the sensor used to capture the human body posture has the characteristics of small size, high sensitivity, and is easy for users to carry [19, 20].

Abobakr et al. proposed a holistic posture-based analysis model [21] that uses the Kinect. The sensor acquires the data, estimates the joint angle of the human body by inputting the depth image and uses a deep convolutional neural network model for the joint perspectives for regression [22], uses comprehensive training images to simulate different body movement tasks, and obtains highly generalized learning models to achieve higher attitude prediction rate [23]. In 2019, Xu et al. implemented depth information and skeletal tracking based on Microsoft Kinect V2 sensors to perform human posture recognition [24], and based on this, human fall detection was implemented. First, a Kinect V2 sensor was used to process the human joint data generated by the skeletal tracker, and then, the optimized BP neural network is used for posture recognition and based on this to detect falls, by training the neural network using a dataset generated by the Kinect tracker and using other body trackers for testing. Finally, posture recognition and fall detection were experimentally validated and tested in real time over the entire operating range of the sensor. The overall accuracy of the NITE tracker used for the drop test was experimentally 98.5%, and the worst accuracy was 97.3 percent. University of Brahem et al. mounted an accelerometer on the foot to track and identify foot movements [25]. University of Schwarz et al. used a MEMS sensor to capture and recognize hand movements, which in turn accomplished a medical office doctors' human-computer operation with a computer

[26]. The feedback from the sensors effectively reduces the possibility of injury during jumping [27]. Lim et al. at Nanyang Technological University, Singapore, invented a wearable wireless human arm motion capture sensing system [28] that captures and recognizes human posture using acceleration sensors and bending sensors for human-computer interaction in medical applications for stroke patients in recovery training. Wang et al. analyzed the signal characteristics of accelerometers and gyroscopes on representative [29], the feature information is extracted, a DT model-based classifier is proposed, and the angle deviation is weighted by an improved PCA algorithm. On average, the experimental results proved that the average accuracy of the pose of other was close to 97.1%, improving the PCA-based angular bias method judgment accuracy.

In 2018, Cai et al. presented a process analysis and Fisher vector-based encoded human action recognition framework [30]; first, by applying Procrustes analysis and local retention projections, apply pose-based features extracted from silhouette images. The distinguishing shape information and the local manifold structure of the human pose are preserved and remain invariant for translation, rotation, and scaling. After the pose features are extracted, a recognition framework based on Fisher vector coding and multiclass support vector machines is used for the human motion classification, and the experimental results demonstrated the effectiveness of the method.

3. Control Strategies

The FES controller is generally composed of a series of preset states and state transition conditions, input signals, and output functions. In this case, each "preset state" corresponds to a movement phase, and the "output function" of each state performs a gradual change of muscle stimulation to its respective targets (the target can be zero) and finally maintains it on these targets. "Condition of state transition" means the precondition of exiting each movement stage. The potential "input signal" set of FSM controller can be the data measured by accelerometer units connected to different parts of the body, angle data, button status, and clock time.

3.1. FSM Controller for Upper-Limb FES. The general existence form of FSM controller is composed of a series of movement phases with time sequence (real rectangle) and natural transition (solid arrow) between each movement phase. Figure 1 shows the transition between states. There is a neutral phase in FSM, that is, the first state, which does not involve in stimulating any muscle parts. Users can customize the total number of stages of FSM controller, but not less than 2 states, which is determined by the selected execution task. FSM returns to the first phase, the neutral phase, whenever the transition conditions of the final phase are met. Therefore, the execution of functional tasks is always in the neutral state. Dashed arrows indicate transitions between special phases (such as default timeout and emergency stop), and any phase can transition to a neutral phase. It is specified that normal transition has higher priority than abnormal transition.

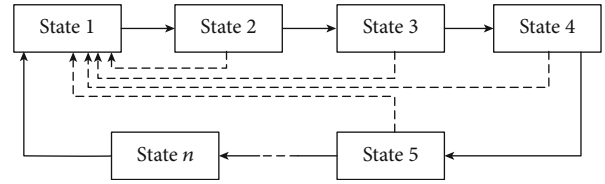


Figure 1: The transition between each state.

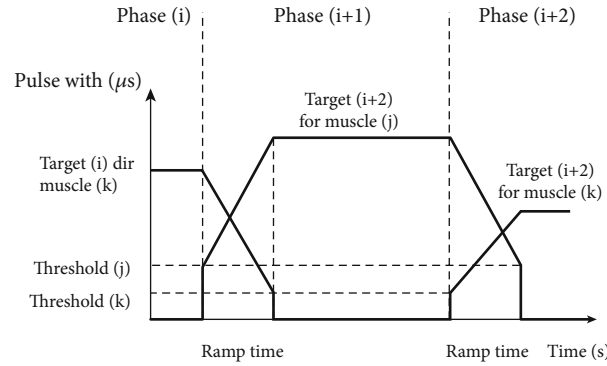


Figure 2: Rise to a threshold (before the rise) and fall from a threshold (after the ramp down).

The timing of the transition between phases is determined by the condition of the state transition, and the timing of the transition is expressed by the input signal and the current state. GUI can be used to customize the parameters of FSM, including the number of states, stimulus parameters of each state (stimulation thresholds, ramps, and target), and state transition conditions (timeout, combination logic, angle triggers, etc.).

Ramp time is a user-defined parameter in FES, which represents the ramp time from the current target to the new target. Figure 2 illustrates the variation of pulse width. The ramp rate is determined by the ramp time and two consecutive nodes in the stimulus curve. In this way, when the stimulus level is different, the shorter the ramp time, the higher the ramp rate.

The realization of ramp is determined by the frequency of FSM. In this paper, we decide to use 20 Hz; then, the minimum time step is 0.05 seconds. This prevents users from noticing any delay.

Phase conversion is determined by the current input signal and conversion conditions. The FSM controller can obtain up to four acceleration data to capture the motion of each part of the upper limb (i.e., hand, upper arm, and lower arm). In this case, the acceleration data of accelerometer will be transmitted to FSM controller in real time during the execution of functional tasks. In order to improve the flexibility of the system, this paper uses logical operators (N/A and OR) to combine two Boolean conditions to create conversion rules.

Let us discuss an example to show how FSM can customize settings for specific FES tasks. As shown in Figure 3, this FES task consists of five phases: "neutral," "reach for door," and "grass handle." In stage 5, the forearm extensor is stimulated to reach the state of releasing the door handle. The

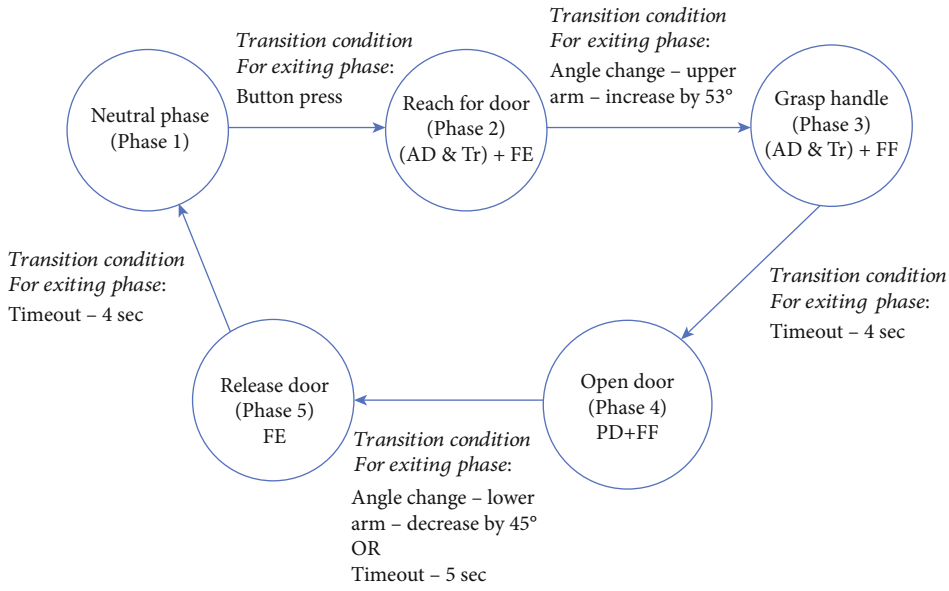


Figure 3: Transition between phases.

transition between states is an instantaneous event after the conditions are met. This example will be carried out in the experimental part, and the realization of each part of FSM is described in detail.

The execution of FES task is reflected in the FSM controller, which can be regarded as the state transition of each movement stage according to a certain time sequence and transition conditions. Table 1 lists the Boolean conditions of each phase transition in the door opening task. Two accelerometers are used to record the movements of the lower arm and the upper arm, respectively, which can be used as the transition of phases 2 to 3 and 4 to 5, and it can also be used as a condition for triggering phase transition. In the transition from stage 4 to stage 5 in this example, the logical operator is OR, which means that only one of condition A and condition B is satisfied and the phase transition be carried out, that is, the upper arm angle is reduced by 45° and the phase 4 is kept for 5 seconds, and the state transition can be triggered. It should be noted that the transition between phases depends not only on the state transition conditions, but also on the current state.

3.2. Implementation of the Finite State Machine Controller. This paper uses MATLAB and Simulink to implement a real-time FSM controller under the Windows platform. The real-time online data acquisition, processing, and stimulation parameter control are realized by Simulink. The components and input/output of the FES control system are described in Figure 4. The FSM controller can input the button pressing signal in real time, time-out clock and three-axis acceleration data, and real-time output of stimulus pulse width (μsec), pulse amplitude (MA), and waveform. The waveform is preset and fixed; the Simulink simulation system runs at 20 Hz, implementing real-time angle tracking, angle triggering, FSM controller and security review.

Table 1: Transition rules of the "open a door" task.

Transition between phases	LO (logical operator)	Factor A	Factor B
1 to 2	Not/and	Press the space bar	Invalid
2 to 3	Not/and	Increase upper arm by 53°	Invalid
3 to 4	Not/and	Hold for 4 sec	Invalid
4 to 5	OR	Decrease upper arm by 45°	Hold for 5 sec
5 to 1	Not/and	Hold for 4 sec	Invalid

The real-time input of FSM controller includes the absolute angle value of x-axis and vertical direction measured by Xsens unit, the "space bar" button in GUI is used as the switch state button, the "enter" key is used as the emergency state button, and the time-out clock time is also included.

The design of FSM controller includes the design of state transition control, stimulation output control, and the research of improving the robustness of angle trigger. The output of FSM will be transmitted to the safety module in real time. The safety module is located between the controller and the stimulator. The safety block can prevent the pain caused by improper stimulation level. Because the safety block will limit the pulse width of a single pulse, the pulse amplitude, i.e., the total charge, and the maximum step size of the ramp, the safety block will stop the stimulation of the stimulator when any limit is exceeded to verify the security of the whole system. Figure 5 is the flow chart of FES control system of upper limb.

4. A Proposed Posture Recognition Algorithm

4.1. Data Acquisition Equipment. In this paper, the MPU6050 sensor module that satisfies the above

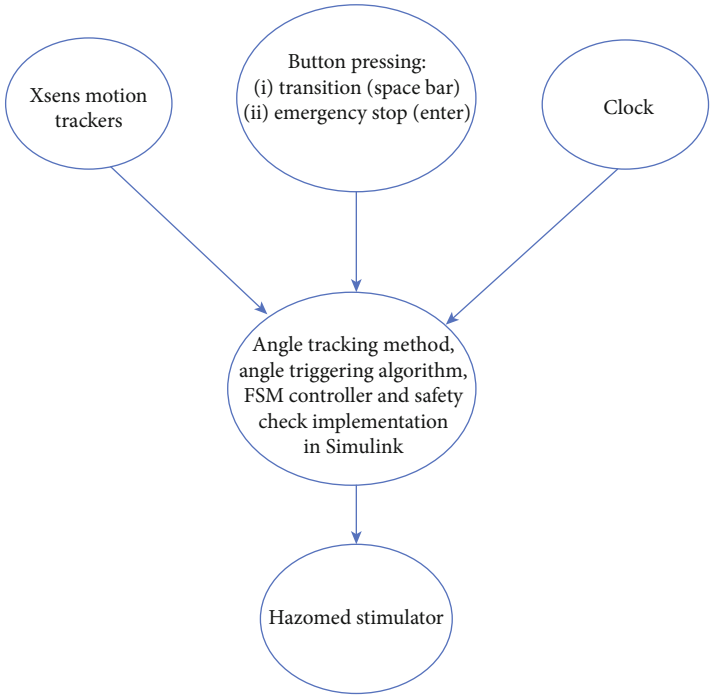


Figure 4: The data flow in FES control system.

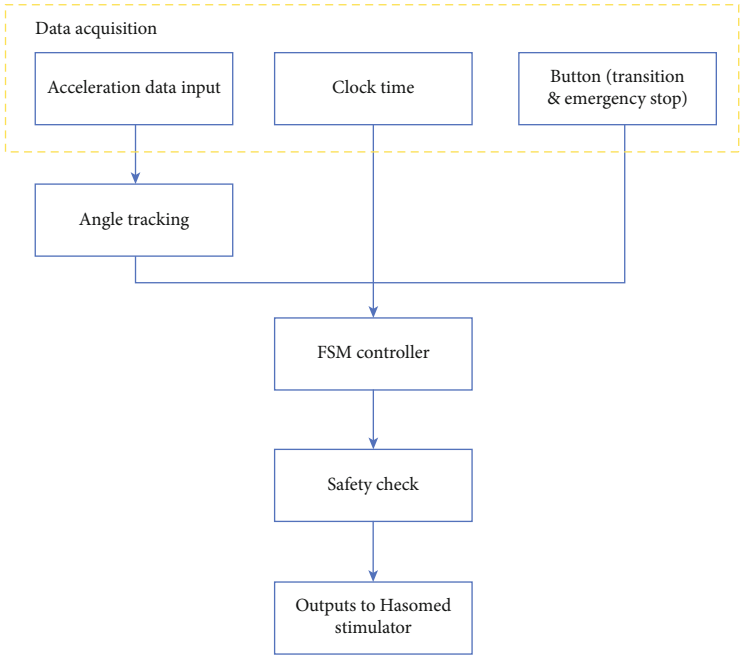


Figure 5: The flow chart of upper-limb FES control system.

characteristics is used as a data acquisition device to provide a reliable data source for subsequent research work.

The MPU6050 is a scalable digital motion sensor that integrates a 3-axis MEMS accelerometer and a 3-axis MEMS gyroscope processor, which accurately tracks fast and slow movements. The data collection device is shown in Figure 6. The measurement range of the sensor is user-definable, and the accelerometer can sense ranges of ±2 g, ±4 g, ± 8 g, and ± 16 g. The angular velocity can be sensed in the

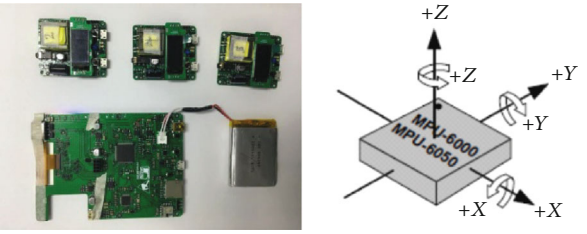


Figure 6: The data collection device.

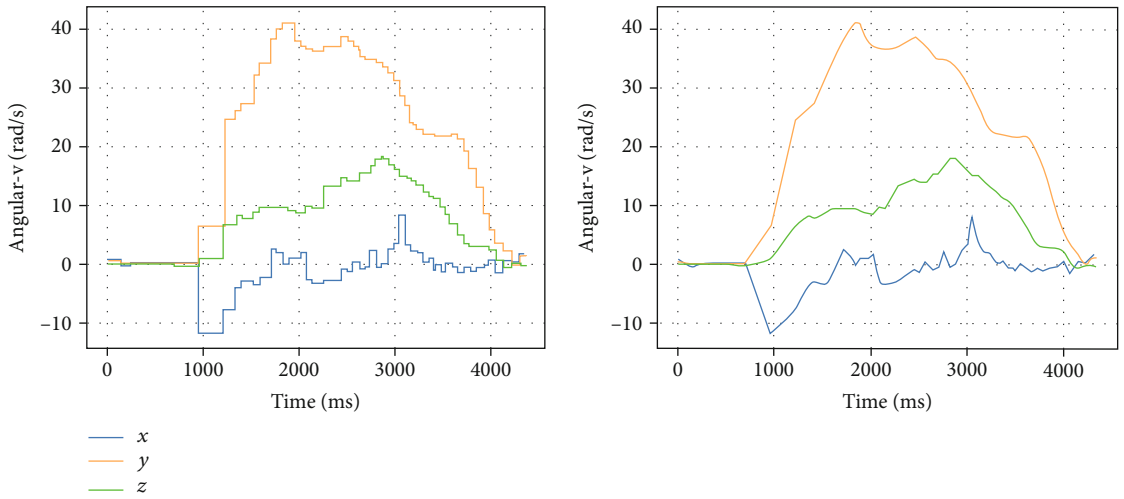


Figure 7: Sensor data waveforms before and after weight removal.

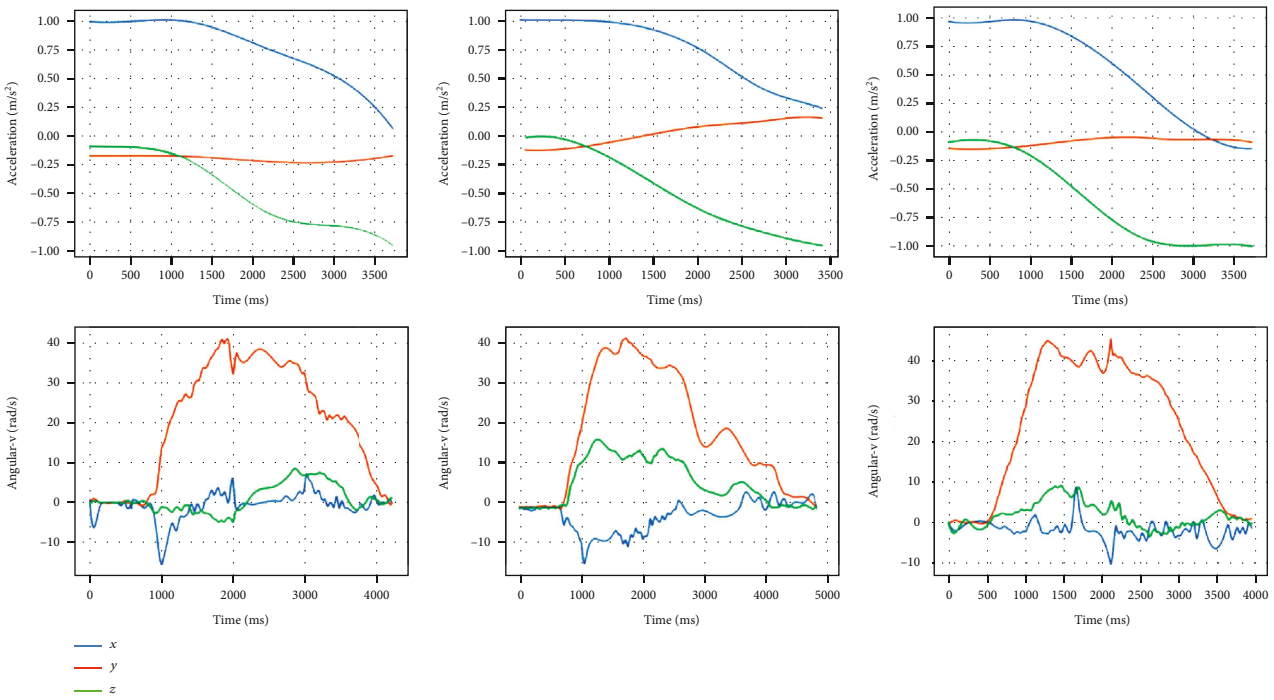


Figure 8: Waveform diagram after preprocessing of the side lift data.

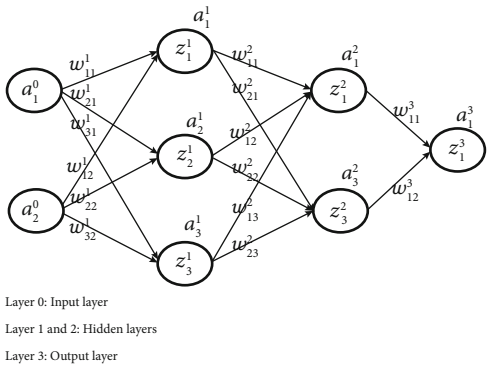


Figure 9: Structural diagram of a fully connected neural network.

range of ±250, ±500, ±1000, and ±2000°/sec (dps). In the data acquisition process, the MPU6050 first puts the calculated values into registers, and then, the microcontroller reads them via I2C.

4.2. Data Preprocessing. To further process the raw dataset, the dataset was deweighted, using the gyroscope data as an example, and the waveforms before and after deweighting are shown in Figure 7.

Data sawtooth has been eliminated, but still not smooth enough, in order to complete part of the missing value, the need to interpolate the dataset to get a smoother interpolation function and the use of three-sample interpolation on

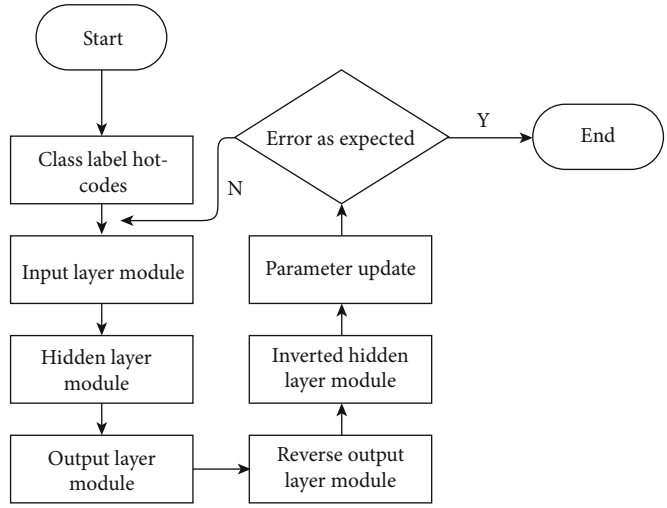


Figure 10: Posture recognition schemes for fully connected neural networks.

the dataset to deal with the processing of A, B, C three sensors of the attitude signal shown in Figure 8.

4.3. Fully Connected Neural Network Model. The experiments are mainly conducted using time domain analysis for feature extraction, with N denoting the number of rows of data in a time window and i denoting the row of data, and the selected variance, range, and interquartile range as features define as follows:

$$\sigma_X^2(t) = \lim_{n\to\infty} \frac{1}{N} \sum_{i=1}^{N} [X_i(t) - \mu_X(t)]^2 = E\left[X^2(t)\right] - (E[X(t)])^2,$$
$$R = X_{\max} - X_{\min},$$
$$\text{IQR} = Q^{-}3 - Q^{-}1. \tag{1}$$

The simple structure of a fully connected neural network is shown in Figure 9,

where a_i^l denotes the output of the neuron, where l denotes the number of layers and i denotes the neuron number; z_i^l denotes the output of the inactivated neuron, where l denotes the number of layers and i denotes the neuron number; w_{ij}^l denotes the weighting factor of the neuron. The fully connected neural network obtains the output as the input of the next layer neuron through the multiplication and accumulation of the input data and the weight and then calculates the activation function to realize the forward propagation calculation. According to the error between the final output layer result and the expected result, the weight parameters are adjusted by the back propagation algorithm until the error between the output and the expected result are acceptable.

The experimental posture recognition scheme based on a fully connected neural network is shown in Figure 10.

Hot codes are performed on the labels of the posture dataset to convert the label variables into a form that the neural network can easily exploit to model operational efficiency as well as the nonlinear capabilities of the model.

Table 2: Stimulation parameters for each channel at different phases.

Phase / Channel	1	2	3	4	5
1	0	108	108	0	0
2	0	54	0	0	72
3	0	0	72	72	0
4	0	0	0	90	0

A fully connected neural network model is constructed. The fully connected network model constructed in this paper consists of four components. The first one is the input layer module, which is responsible for inputting the format of the posture data and the initialization task of neuron parameters at each layer during the first execution, setting for each reading of a set of 1590×6 pose matrix data. The hidden layer module consists of a hidden layer containing 30 neurons, the number of layers is determined by comparing the recognition rate and is responsible for the upper layer neurons, the output data are weighted and summed, and the activation function is used to generate the input values from the lower layer neurons. The output layer module is responsible for obtaining the predicted probability values for the six postures from the incoming data from the upper layer neurons. The tuning module is responsible for calculating the activation value for each neuron, the loss of each layer based on the activation value, and the parameter gradient from the output of the layers start to make parameter adjustments going forward. The posture dataset is trained by the above method to derive the final recognition model.

5. Experiment

5.1. FES Control Test. This paper uses the "open door" task to test the FSM controller. The output data of the main monitoring controller includes the following: the accelerometer

Table 3: Stimulated muscles and their stimulation changes at various stages.

Phase	Output
Neutral	No stimulation
Reach for door	Anterior deltoid, triceps, and forearm extensors ramp from threshold to target and then stay on the targets.
Grasp	Both triceps and anterior deltoid rise to the target. Forearm flexors go from threshold to target. Both channels are at the target location. Forearm extensors shut down by climbing to a threshold and then decreasing to zero.
Open door	Forearm flexors ramp towards the next target. Posterior deltoid goes from threshold to target. Both channels stay at their target location. Anterior deltoid and triceps turn off by ramping to threshold and then decreasing to zero.
Release	Forearm extensors ramp from threshold to target and stay at the target. Posterior deltoid and forearm flexors turn off by ramping to threshold and then decreasing to zero.

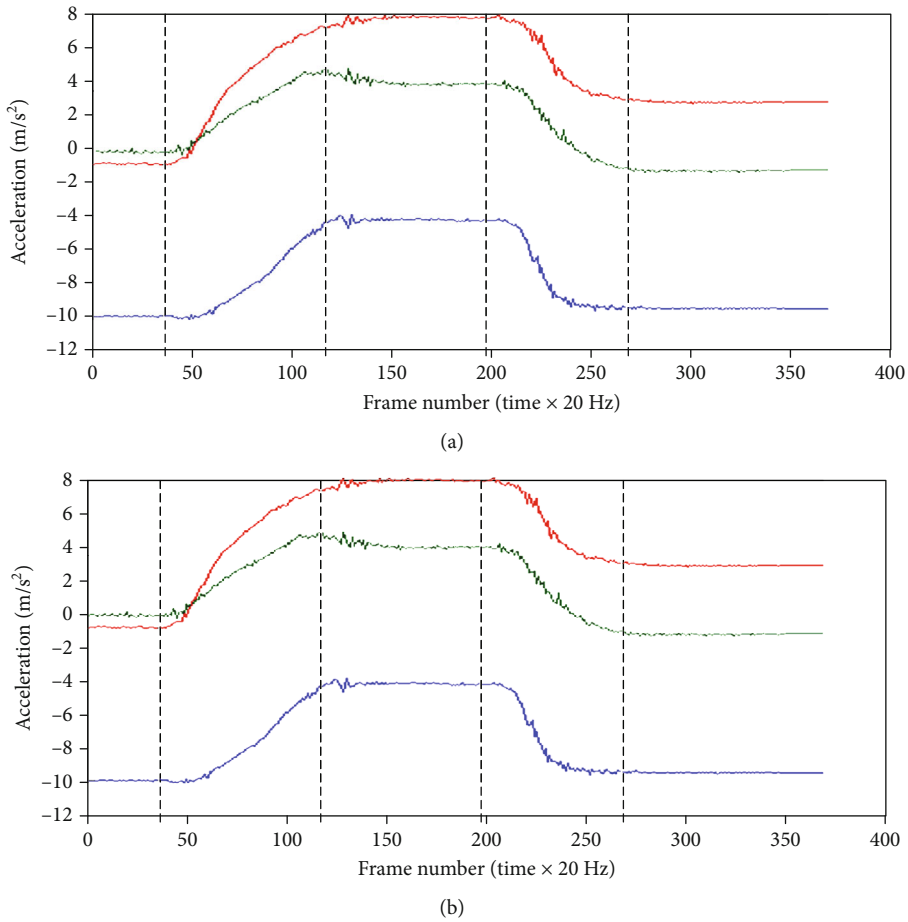


Figure 11: Acceleration data. (a) Upper arm and (b) forearm.

signal of the Xsens unit of the upper arm and forearm, the change of the vertical angle after each state conversion, and the phase number and pulse width of each part of the muscle.

Before running FSM controller, it is necessary to install the Xsens motion tracking software. The software can directly collect the real-time acceleration data of Xsens inertial sensor unit on MTX hub from MATLAB. Xsens system samples the sensor data at the frequency of 100 Hz. Refer to Figure 3 for the muscle parts and transition conditions involved in the transition of each stage. The specific stimulation parameters can be seen in Tables 2 and 3. The "stimulus threshold" and "maximum comfortable stimulation" are the default values, which are 360 μs and 0 μs, respectively.

The data is collected by healthy subjects in real time when executing the "open a door" task. The dotted lines in the figure below indicate the transition between states. Two Xsens are, respectively, installed on the forearm and the upper arm to trigger at the starting angle. The corresponding acceleration data is shown in Figure 11.

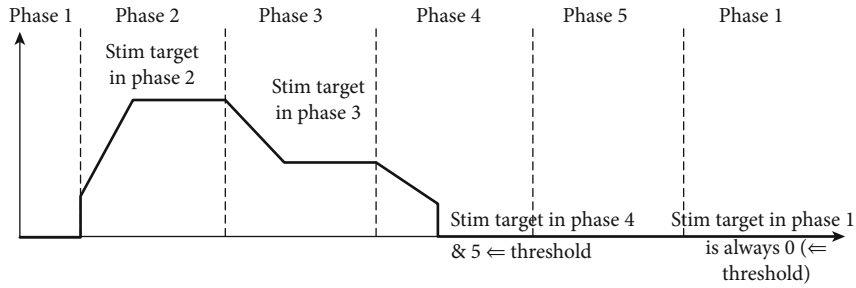


Figure 12: The stimulating curve of anterior deltoid and triceps at different phases.

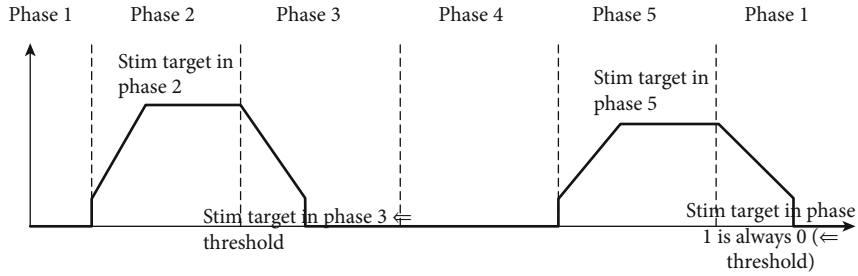


Figure 13: The stimulation curve of forearm extensors at each phase.

Figures 12 and 13 show the stimulation of part of the muscle tissue at all phases of movement in the example task of "opening the door" (see Figure 3). The stimulated muscles corresponding to each stage will ramp to the target level of that phase.

5.2. Neural Network Test. In order to verify the effectiveness of the fully connected neural network model for human posture recognition, this section takes the six human posture data collected above as an example and performs experimental validation.

The experimental dataset contains the six classical postures of forward flattening, lateral flattening, upward elbow bending, bent elbow backward, wrist upward bending, and horizontal elbow flexion MEMS sensor signals; in order for the pose dataset to be applied to the neural network model, the dataset needs to be preprocessed first. Since the completion time required for various postures varies, the length of the sensor signals collected for the posture samples is inconsistent, so as not to lose the original information of the attitude, it requires adding the original signal data to make the data window consistent. Before performing the experiments, this paper starts with a procedure to find the longest pose sample for the prelift, with a completion time of 5.3 seconds, and to add all the sensor data through three-sample interpolation for the dataset plus windows, interpolation is complete, splicing three sensor data, so that each attitude of the data sample then becomes $1590*6$ in the form of a two-dimensional matrix. When solving multiclassification problems using neural networks, the labels need to be digitized, and the digitized class labels converted to binary matrix representation, such an operation is called creating dummy variables (one hot encoding) from categorical variables. As an example, the anterior flattened pose data used in this paper is transformed into the following labels: [0, 1, 0, 0, 0, 0].

Table 4: Identification results for different numbers of hidden layers.

Number of hidden layers	Average recognition rate	Time (seconds)
1	91.27%	14.576
2	81.34%	18.743
3	94.08%	20.492
4	84.18%	26.533
5	89.87%	23.33
6	79.41%	31.106

Table 5: Identification results for different activation functions.

Activation function	Relu	Softplus	Sigmoid	Tanh	Softsign
Average accuracy	91.31%	93.07%	37.84%	81.96%	90.33%

In order to study the effect of the number of hidden layers on the recognition accuracy and recognition efficiency, this paper investigates the recognition accuracy of hidden layers 1 to 6 and the time taken; the number of neurons was all 30, and the judgment index was the recognition accuracy. The comparison results are shown in Table 4. In addition, the effects of different activation functions and optimizers on recognition accuracy are compared. The experimental results are shown in Tables 5 and 6.

To summarize the above comparative experiments, the fully connected neural network selected a 3-layer hidden layer structure with an activation function of softplus as well

Table 6: Identification results of different optimizers.

Optimizer	adam	rmsprop	sgd	adadelta	adagrad	adamax
Average accuracy	94.25%	96.01%	13.07%	93.17%	97.19%	94.35%

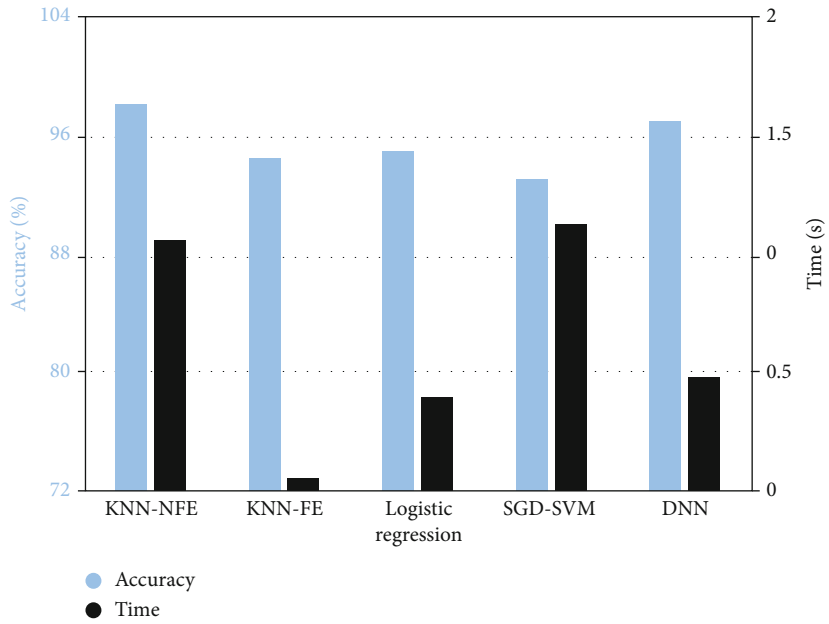


Figure 14: Accuracy and computation time of each algorithm (blue represents the accuracy, and black represents the time).

as an adaptive gradient descent optimizer. And three ten-fold cross-validation to take the mean value, the recognition rate, and duration of each algorithm are calculated as shown in Figure 14.

Known datasets without feature extraction retain good pose information, and the KNN model has a very good handle on such pose datasets, good recognition performance (KNN-NFE) with up to 98% recognition accuracy. However, due to the large sensor signal data, the resulting computation time is costly and takes as much as 1 second. In contrast, the calculation time of KNN classifier after feature extraction has been shortened by an order of magnitude and improved greatly, but due to the pose information was incomplete and the average recognition rate dropped to 94%. The logistic regression model outperformed the stochastic gradient descent SGD using a linear support vector machine classifier in terms of recognition rate and computation time classifier, and the recognition rate is also improved compared to the feature extracted KNN model. In addition, the fully connected neural network model has a similar recognition rate and takes less time to compute than the KNN-NFE, which has the highest recognition rate. Therefore, combining the recognition accuracy and time efficiency, fully connected neural networks still have some superiority in pose recognition.

FES control experiment designed the corresponding finite state machine control strategy for the door opening task. The experiment involved the muscle stimulation site, stimulation parameters, the transition of each stage, and the transition between each state. The experimental results proved the effectiveness of the control strategy of finite state machine, which provided a powerful solution for the clinical design of rehabilitation plan and the implementation of rehabilitation training. The experiment of posture recognition is based on the recognition of six basic upper-limb movements. By comparing different classification methods, the practicability of fully connected neural network in posture recognition is finally determined. This discovery can be combined with the rehabilitation evaluation of patients in the later stage and can be used as a reference basis for the evaluation of patients' rehabilitation degree, which is of great significance to patients' rehabilitation.

6. Conclusions

This paper proposes a FSM controller model that supports clinical users to personalize settings according to different FES upper-limb functional tasks, which can be used as a powerful tool for clinicians to customize treatment plans for patients with different degrees of nerve injury. The implemented FSM controller was tested through the "door opening" task, and the experimental results proved its effectiveness and feasibility. The model is flexible and convenient, which greatly improves the convenience of the rehabilitation system for patients with upper-limb stroke.

In order to identify human posture, this paper starts with building a posture data acquisition platform and collects 6 of them in a three-channel data acquisition mode. The classical posture is recorded in the MEMS sensor data. Then, preprocessing such as deweighting and triple sample bar

interpolation was applied to the acquired dataset, and time domain analysis was applied from the sensor signal. Features useful for posture recognition are extracted. Subsequently, KNN, logistic regression, and random gradient descent were performed using an experimentally validated classification model with the goal of recognizing human posture experiments of the algorithms. To verify the superiority of each algorithm, the data window was adjusted to compare the recognition speed, computation duration, and accuracy of each classifier. In order to improve the accuracy of human posture recognition, a fully connected neural network-based model is established. In the process of constructing the network model, this paper investigates different activation functions and optimizers, and after experimental comparative analysis, it selects the better-performing softplus activation function as well as adagrad optimizer. Finally, by comparing the combined recognition accuracy and time efficiency with other classification models, the adjusted fully connected neural model in human is more effective and superior in posture recognition.

In this paper, based on small sample data, we establish a high-precision attitude recognition model, but there is still room for improvement, especially for the problem that the effect of small sample data in deep learning model is not as good as large-scale data. In the future, we will try to study a kind of attitude data that can generate typical attitude data by learning the characteristics of attitude data, so as to achieve the effect of expanding the sample data, and further improve in solving the problem of insufficient training samples.

Data Availability

The dataset is prepared by three-axis acceleration prototype nodes developed by ourselves for collection.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Xian Yu and Bo Xiao are the first authors due to equal contribution to this paper.

Acknowledgments

This work has received funding from Science and Technology Project of SGCC Research on Feature Recognition and Prediction of Typical Ice and Wind Disaster for Transmission Lines Based on Small Sample Machine Learning Method, National Natural Science Foundation of China (Nos. 41911530242 and 41975142), 5150 Spring Specialists (05492018012 and 05762018039), Major Program of the National Social Science Fund of China (Grant No. 17ZDA092), 333 High-Level Talent Cultivation Project of Jiangsu Province (BRA2018332), Royal Society of Edinburgh, UK and China Natural Science Foundation Council (RSE Reference: 62967_Liu_2018_2) under their Joint International Projects funding scheme, National Natural Science Foundation of China (Grant No. 41875184), Innovation Team of "Six Talent Peaks" in Jiangsu Province (Grant No. TD-XYDXX-004), and basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20191398 and BK20180794).

References

[1] J. D. Pandian, S. L. Gall, M. P. Kate et al., "Prevention of stroke: a global perspective," *The Lancet*, vol. 392, no. 10154, pp. 1269–1278, 2018.

[2] M. Ferrarin, F. Palazzo, R. Riener, and J. Quintern, "Model-based control of FES-induced single joint movements," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 9, no. 3, pp. 245–257, 2001.

[3] J. J. Daly, E. B. Marsolais, L. M. Mendell, W. Z. Rymer, and A. Stefanovska, "Therapeutic neural effects of electrical stimulation," *IEEE Transactions on Rehabilitation Engineering*, vol. 4, no. 4, pp. 218–230, 1996.

[4] G. You and T. Yan, "Functional electrical stimulation and its application in post-stroke hemiplegic patients," *Chinese Journal of Physical Medicine and Rehabilitation*, pp. 142–144, 2007.

[5] W. Liberson, H. Holmquest, D. Scot, and M. Dow, "Functional electrotherapy: stimulation of the peroneal nerve synchronized with the swing phase of the gait of hemiplegic patients," *Archives of Physical Medicine and Rehabilitation*, vol. 42, pp. 101–105, 1961.

[6] S. K. Sabut, P. K. Lenka, R. Kumar, and M. Mahadevappa, "Effect of functional electrical stimulation on the effort and walking speed, surface electromyography activity, and metabolic responses in stroke subjects," *Journal of Electromyography and Kinesiology*, vol. 20, no. 6, pp. 1170–1177, 2010.

[7] S. Obuz, V. H. Duenas, R. J. Downey, J. R. Klotz, and W. E. Dixon, "Closed-loop neuromuscular electrical stimulation method provides robustness to unknown time-varying input delay in muscle dynamics," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2482–2489, 2020.

[8] M. Lin, B. Porr, and H. Gollee, "Technical advances in functional electrical stimulation in gait function recovery," *Instrumentation Journal*, vol. 38, no. 6, pp. 1319–1334, 2017.

[9] Q. Zhang, M. Hayashibe, and C. Azevedo-Coste, "Evoked electromyography-based closed-loop torque control in functional electrical stimulation," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 8, pp. 2299–2307, 2013.

[10] Y. Zhou, Y. Fang, K. Gui, K. Li, D. Zhang, and H. Liu, "sEMG bias-driven functional electrical stimulation system for upper-limb stroke rehabilitation," *IEEE Sensors Journal*, vol. 18, no. 16, pp. 6812–6821, 2018.

[11] A. Dodson, N. Alibeji, and N. Sharma, "Experimental demonstration of a delay compensating controller in a hybrid walking neuroprosthesis," in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 465–468, Shanghai, 2017.

[12] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines-a survey," *Proceedings of the IEEE*, vol. 84, no. 8, pp. 1090–1123, 1996.

[13] Y. Wang, B. Metcalfe, Y. Zhao, and D. Zhang, "An assistive system for upper limb motion combining functional electrical

stimulation and robotic exoskeleton," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 2, pp. 260–268, 2020.

[14] S. M. Cerqueira, L. Moreira, L. Alpoim, A. Siva, and C. P. Santos, "An inertial data-based upper body posture recognition tool: a machine learning study approach," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 4–9, Ponta Delgada, Portugal, 2020.

[15] Y. Xi, Y. Zhang, S. Ding, and S. Wan, "Visual question answering model based on visual relationship detection," *Signal Processing: Image Communication*, vol. 80, article 115648, 2020.

[16] Z. Zhang, Y. Liu, A. Li, and M. Wang, "A novel method for user-defined human posture recognition using Kinect," in *2014 7th International Congress on Image and Signal Processing*, pp. 736–740, Dalian, China, 2014.

[17] S. Neili, S. Gazzah, M. A. El Yacoubi, and N. E. Ben Amara, "Human posture recognition approach based on ConvNets and SVM classifier," in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–6, Fez, Morocco, 2017.

[18] Y. Chen, L. Yu, K. Ota, and M. Dong, "Hierarchical posture representation for robust action recognition," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1115–1125, 2019.

[19] G. Orengo, A. Lagati, and G. Saggio, "Modeling wearable bend sensor behavior for human motion capture," *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2307–2316, 2014.

[20] Y. Yan and Y. Ou, "Accurate fall detection by nine-axis IMU sensor," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 854–859, Macau, Macao, 2017.

[21] A. Abobakr, D. Nahavandi, and J. Iskander, "RGB-D human posture analysis for ergonomie studies using deep convolutional neural network," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2885–2890, Banff, AB, Canada, 2017.

[22] Y. Zhao, H. Li, S. Wan et al., "Knowledge-aided convolutional neural network for small organ segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1363–1373, 2019.

[23] L. Wang, H. Zhen, X. Fang, S. Wan, W. Ding, and Y. Guo, "A unified two-parallel-branch deep neural network for joint gland contour and segmentation learning," *Future Generation Computer Systems*, vol. 100, pp. 316–324, 2019.

[24] Y. Xu, J. Chen, Q. Yang, and Q. Guo, "Human posture recognition and fall detection using Kinect V2 camera," in *2019 Chinese Control Conference (CCC)*, pp. 8488–8493, Guangzhou, China, 2019.

[25] M. B. Brahem, B. A. J. Ménélas, and M. J. D. Otis, "Use of a 3DOF accelerometer for foot tracking and gesture recognition in mobile HCI," *Procedia Computer Science*, vol. 19, pp. 453–460, 2013.

[26] L. A. Schwarz, A. Bigdelou, and N. Navab, "Learning gestures for customizable human-computer interaction in the operating room," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 129–136, Toronto, ON, Canada, 2011.

[27] A. V. Dowling, J. Favre, and T. P. Andriacchi, "Inertial sensor-based feedback can reduce key risk metrics for anterior cruciate ligament injury during jump landings," *The American Journal of Sports Medicine*, vol. 40, no. 5, pp. 1075–1083, 2012.

[28] C. K. Lim, Z. Luo, I. M. Chen, and S. H. Yeo, "Wearable wireless sensing system for capturing human arm motion," *Sensors and Actuators A: Physical*, vol. 166, no. 1, pp. 125–132, 2011.

[29] B. Wang, X. Liu, B. Yu, R. Jia, and L. Huang, "Posture recognition and heading estimation based on machine learning using MEMS sensors," in *International Conference on Artificial Intelligence for Communications and Networks*, pp. 496–508, Harbin, China, 2019.

[30] J. X. Cai, R. X. Zhong, and J. J. Li, "Silhouettes based human action recognition in video via procrustes analysis and fisher vector oding," *Journal of Donghua University*, vol. 2, pp. 4–9, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6611867, 12 pages
https://doi.org/10.1155/2021/6611867

Research Article

Enhancing Dynamic Binary Translation in Mobile Computing by Leveraging Polyhedral Optimization

Mingliang Li, Jianmin Pang, Feng Yue, Fudong Liu, Jun Wang, and Jie Tan

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, Henan 450000, China

Correspondence should be addressed to Jianmin Pang; jianmin_pang@hotmail.com

Received 27 November 2020; Revised 9 March 2021; Accepted 5 April 2021; Published 19 April 2021

Academic Editor: Shaohua Wan

Dynamic binary translation (DBT) is gaining importance in mobile computing. Mobile Edge Computing (MEC) augments mobile devices with powerful servers, whereas edge servers and smartphones are usually based on heterogeneous architecture. To leverage high-performance resources on servers, code offloading is an ideal approach that relies on DBT. In addition, mobile devices equipped with multicore processors and GPU are becoming ubiquitous. Migrating x86_64 application binaries to mobile devices by using DBT can also make a contribution to providing various mobile applications, e.g., multimedia applications. However, the translation efficiency and overall performance of DBT for application migration are not satisfactory, because of runtime overhead and low quality of the translated code. Meanwhile, traditional DBT systems do not fully exploit the computational resources provided by multicore processors, especially when translating sequential guest applications. In this work, we focus on leveraging ubiquitous multicore processors to improve DBT performance by parallelizing sequential applications during translation. For that, we propose LLPEMU, a DBT framework that combines binary translation with polyhedral optimization. We investigate the obstacles of adapting existing polyhedral optimization in compilers to DBT and present a feasible method to overcome these issues. In addition, LLPEMU adopts static-dynamic combination to ensure that sequential binaries are parallelized while incurring low runtime overhead. Our evaluation results show that LLPEMU outperforms QEMU significantly on the PolyBench benchmark.

1. Introduction

A DBT system can run guest applications transparently on a host machine with a different Instruction Set Architecture (ISA), and DBT is gaining importance in mobile computing [1]. Firstly, smartphones are commonly used for various purposes in daily life. With rapid advancements in mobile computing, mobile devices equipped with multicore processors and GPUs are becoming ubiquitous. Powerful computation resources allow a wide range of x86_64 application binaries to be migrated to mobile devices transparently using DBT, e.g., multimedia and image applications. Mobile Edge Computing (MEC) augments mobile devices with powerful servers and provides mobile devices with opportunities to run computation-intensive applications by leveraging edge-based computational resources [2]. However, edge-based servers and smartphones typically adopt heterogeneous architecture, while code offloading which also relies on DBT is an ideal approach for leveraging high-performance resources on servers [3].

However, concerns about performance and translation efficiency constrain the use of DBT in mobile computing. There are two major factors that affect DBT performance: (1) translation and emulation overhead and (2) translated code quality. A DBT system translates one section of the guest binary code at a time to the host binary code and then executes it. The translation and emulation overhead of DBT and execution of the translated code together determine the overall performance of a DBT system. To achieve an effective trade-off between the two factors, a DBT system usually performs highly efficient optimizations: it cannot afford sophisticated and complicated optimizations, e.g., polyhedral optimization.

Ubiquitous multicore processors bring us opportunities to improve the performance of DBT. However, traditional DBT systems do not fully exploit the computational

resources provided by multicore processors, especially when translating sequential guest applications. Most studies have focused on leveraging multiple cores to reduce translation and emulation overhead. Approaches such as parallelizing loops in guest binaries and generating concurrent code assigned to multiple cores have been overlooked. One of the key benefits of this method is that DBT can thus achieve significant performance improvement when parallelizing hotspots of guest applications successfully, even outperforming native executing.

In this study, we develop LLPEMU, a DBT framework that can translate and parallelize affine loops in guest application binaries. LLPEMU consists of two components: a dynamic binary translator for nonhotspot translation and emulation and a static translator that extracts and parallelizes affine loops in guest binaries. LLPEMU uses an enhanced QEMU [4] as its dynamic binary translator and Polly [5] as the backend polyhedral optimizer of its static translator. However, instructions for the emulation mechanism in the translated code impede parallelization. We investigate these obstacles and present a feasible way to overcome them. With such a combined static-dynamic approach, we successfully perform polyhedral optimization while ensuring low runtime overhead. Our evaluation results show that LLPEMU outperforms QEMU significantly on PolyBench.

The main contributions of this study are as follows:

(1) We have developed a DBT framework with a combined static-dynamic design that performs polyhedral optimization on affine loops while incurring low runtime overhead

(2) We have proposed a loop-level DBT-specific optimization to provide opportunities to eliminate redundant loads/stores in the target code region

(3) We have investigated obstacles to parallelization that are created by the emulation mechanism of binary translation and presented a feasible method to overcome them

2. Background

As LLPEMU uses an enhanced QEMU as its dynamic binary translator, this section provides background information about QEMU. QEMU [4] is a popular open-source dynamic binary emulator which supports several ISAs such as x86, ARM, MIPS, and PowerPC. In this work, we use the process-level emulation of QEMU in LLPEMU.

QEMU is mainly composed of the two following stages: (1) Emulation: QEMU creates a virtual execution environment to emulate architecture states of virtual guest processors. Starting from parsing guest application executable file, binary code and data are loaded into QEMU memory space. While translating guest applications at the unit of basic block, architecture states of virtual processors stored in memory will be updated according to executing results. (2) Translation: QEMU facilitates quick emulation of multi-ISA by using Tiny Code Generator (TCG) [6]. Figure 1 draws an outline of TCG. The guest machine code on different ISAs is translated into intermediate representation by TCG. In this paper, we use TCG IR to refer to that. After highly efficient IR-level optimization, TCG IR is translated into the host machine code.

The main loop of QEMU is a translation-emulation loop. After QEMU initializes the virtual execution environment, translation starts from entry PC stored in architecture states of virtual processor, which is set during initialization based on information extracted from ELF. The guest code starting from entry PC is translated into TCG IR at the unit of basic block and then converted into the host machine code. The execution of the translated code will alter architecture states, and a new PC will be set. Then, the translation-execution loop restarts from the new PC.

All these works are done at runtime, leading to the conflict of high code quality and low overhead. In fact, QEMU just perform simple optimization liveness analysis and store forwarding on TCG IR. For short-running application, QEMU is an ideal choice. The design of TCG makes QEMU find the good balance between high-quality code and low overhead. Therefore, we choose to develop our binary translation framework based on QEMU.

But when running application with hotspots, especially multimedia applications with loop nests, QEMU does not perform well. Frequent control flow switching between execution of translated host code and QEMU dispatcher results in many load/store instructions to save program context. The cache hit rate has also been greatly reduced.

3. Architecture of LLPEMU

In this section, we first discuss the design issues that arise when developing a DBT framework that parallelizes loop nests in a sequential guest code by leveraging polyhedral optimization. Then, we describe LLPEMU in detail.

3.1. Design Issues. Our goal is to enable a DBT system to parallelize loop nests in the sequential guest code without incurring high runtime overhead. To this end, we focus on the following issues.

First, the runtime overhead due to optimization and parallelization of the target code should be as low as possible. However, analyzing and parallelizing binaries can be expensive. To parallelize the binaries, we must perform complex analysis to recover the CFG and extract the loops. A polyhedral optimizer introduces considerable overhead. Moreover, not all the loops can be parallelized. DBT can benefit from parallel execution only when the code is parallelized.

It is crucial to design the system architecture such that it can extract loops and perform polyhedral optimization without introducing high runtime overhead. On the basis of these considerations, we propose a combined static-dynamic DBT system. A dynamic binary translator is responsible for initializing the virtual execution environment and emulating guest CPUs; loop nests are extracted and parallelized statically ahead of time.

Although some studies have investigated trace formation based on dynamic profiling with acceptable overhead [7], these methods are not entirely suitable for extracting loop

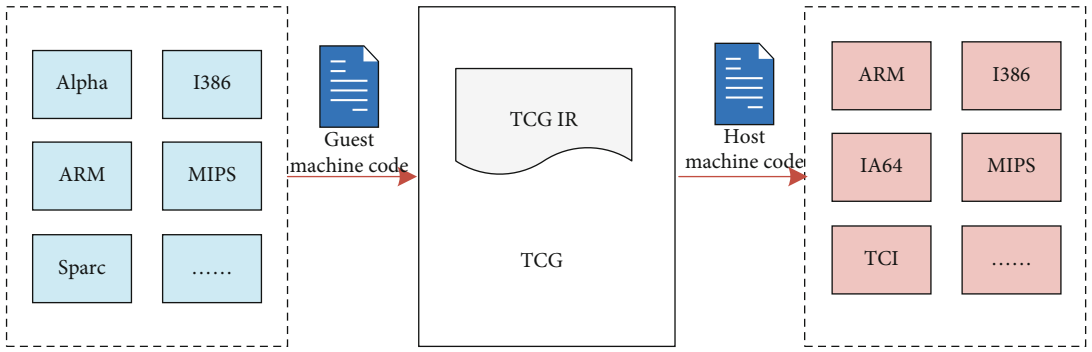


Figure 1: Outline of Tiny Code Generator (TCG).

nests. Trace selection and merging must be performed to format the loop nests. For better selections of traces, trace merging is usually performed on the basis of profiling, which involves high runtime overhead. Moreover, dynamic approaches cannot reduce the overhead of polyhedral optimization.

The second issue is how to parallelize loops extracted from guest binaries. Many polyhedral optimizers have been developed, such as PLUTO [8] and Polly [5]. However, nearly all of them focus on source-level optimization. In this study, we try to adapt these standard compiler tools to a DBT system to parallelize the guest machine code.

Adapting existing compiler methods designed for the source code to the DBT IR is a challenging task. Compiler tools perform analysis and transformations based on the loop structure and symbolic information. Heavy optimization and ISA-specific transformations during compilation make the loop structure and symbolic information unavailable. Thus, lifting the DBT IR up to the source code needs complicated decompilation.

On the basis of these considerations, we choose Polly as our parallelizer. Polly, which has been developed on the basis of LLVM infrastructure [9], analyzes and parallelizes loops at the IR level. Lifting TCG IR to LLVMIR does not require sophisticated decompilation. In addition, there are several general passes for analysis and optimization. We can use these passes to develop our transformation for TCG IR. Furthermore, LLVM can generate a machine code on various ISAs, which matches the retargetability of QEMU. Thus, LLPEMU can also support various host ISAs.

3.2. Overview of Architecture. The architecture and main components of LLPEMU are shown in Figure 2. Considering the adaptation of polyhedral optimization to binary translation with low overhead, LLPEMU consists of two translators.

3.2.1. Static Translator. In the static stage, loops are extracted from the guest application and then translated into the parallelized host machine code. Related information files are also generated to enable the dynamic binary translator to leverage the static analysis results and load the parallelized code. As an offline stage, any complicated analysis and optimization will not increase the runtime overhead. For a specific guest application, loops are extracted and optimized before the first run. However, every run can benefit from a high-quality host code.

3.2.2. Dynamic Translator. The dynamic binary translator is responsible for creating and maintaining the virtual execution environment of the guest application involving memory space mapping and state structure updating. The nonhotspot code is translated and executed in the dynamic stage. If the target PC is the entry of a target loop, the dynamic binary translator will switch the workflow from translation to the execution of a statically generated code.

Next, we describe the details of the two stages and discuss how the static and dynamic stages are combined.

3.3. Static-Dynamic Combination. The static-dynamic combination is central to the LLPEMU in order to achieve an effective trade-off between high code quality and low runtime overhead. It is crucial to determine how the static and dynamic stages can collaborate with each other. We must ensure that the parallelized code generated statically can be executed by DBT when translating the target code. The code switch must not impede the DBT emulation mechanism. In this section, we focus on the following issues.

3.3.1. Maintaining DBT Emulation. As translation and execution proceed, DBT alters both the data in the guest memory and the virtual processor state structure. When translating the target loops, DBT will redirect the control flow from the original translation-execution loop to the execution of the parallelized code generated in the static stage. Then, the translation-execution loop resumes with the end state of the parallelized code.

Therefore, the static translator should not only parallelize the sequential guest binaries but also ensure that the parallelized code it outputs maintains the DBT emulation. DBT can resume translating with the correct state only if the data in the guest memory and the state structure are altered as in the case of DBT.

Many studies have investigated lifting the machine code to the source code partially or to LLVM IR and then parallelizing the lifted code using compiler tools. Through these methods, the sequential guest machine code can be directly converted into the parallelized host code. However, such code cannot maintain the DBT emulation mechanism.

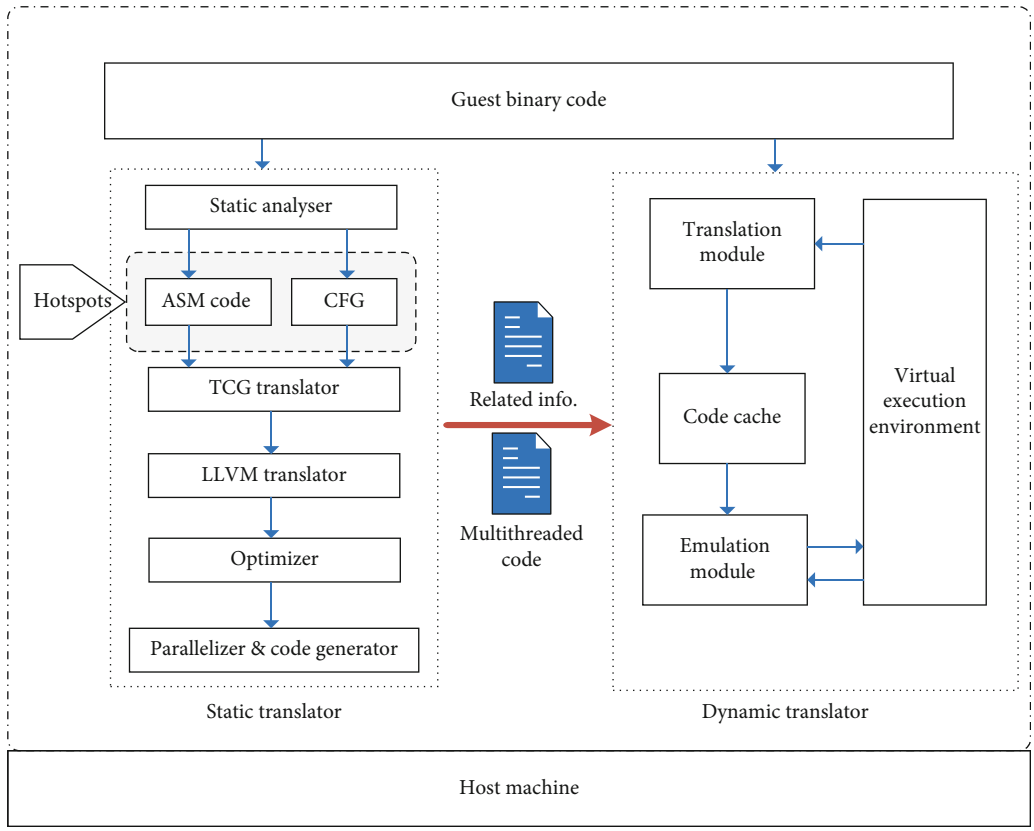


Figure 2: Overview of LLPEMU.

Next, we present our method that makes the static translator output code that is compliant with the DBT emulation mechanism.

First, a static TCG translator is developed to translate the guest machine code into TCG IR before conversion into LLVM IR. As described in Section 2, to emulate the virtual states, TCG will insert memory manipulation instructions to update the state structure in the memory. During execution of the translated code of a basic block, the state structure is altered according to the results of the instructions. Hence, we can exploit the TCG to generate a code with state structure manipulations.

Second, we perform memory space mapping. The base address of the guest application varies with each run. QEMU will choose an appropriate address instead of the address indicated by the ELF file. The target address of memory access to the guest application is computed on the basis of the base address. The actual address of the target memory access is calculated by adding the guest address in the virtual register to the base address. To overcome this obstacle, the base address is set as an argument of the package function; every memory access in packaged loops will be performed on the basis of this argument dynamically.

3.3.2. Interface Design. To leverage the parallelized code generated statically, the dynamic translator must know when and how to switch to the parallelized code. In the static stage, the code is compiled as a shared library using "-O3 -shared -PIC." Each target loop is packaged as a function with the required runtime component pointers as parameters, such as the pointer of the state structure and base address. The function name is a combination of the application name and entry PC of this loop; a specific target loop can be determined from the other loops by the two components.

The dynamic translator loads the shared library using POSIX API during initialization. *Dlopen()* and *dlsym()* are used to obtain a pointer to the function by a given name. Related information about the target loops is read from files to enable the dynamic translator to check whether the succeeding PC is the entry of a target loop. When the dynamic translator reaches the entry of a target loop, it triggers the redirection of the control flow to the execution of the optimized code. The advantage of using POSIX API is that we do not need to relink the optimized code.

3.4. Dynamic Binary Translator. The dynamic binary translator of LLPEMU is developed from QEMU, and additional functionality is added so that it can switch to a parallelized code generated statically when translating target loops. In the case of target loop nests, the control flow of the dynamic translator is switched to the execution of the parallelized code while QEMU translates and emulates the basic blocks one by one. As a result, the context switching overhead is reduced.

During the initialization of the dynamic translator, information about the parallelized loops, including the entry PC of the target loops and function name of the packaged loops, is read by parsing related information files. Based on this

information, the parallelized code is loaded and linked, and the function pointer to the packaged loops is also obtained.

The dynamic translator checks whether the entry PC of a basic block is also the entry of a parallelized loop to trigger the switch of the control flow. A simple approach is to perform a hash-table lookup when translating a new basic block. For a large-scale application with a few parallelized loops, the overhead incurred by a hash-table lookup is unacceptable.

To address this issue, we propose a low-overhead method. The key aspect of our method is avoiding a hash-table lookup by leveraging the code cache of the dynamic translator. As described in Section 2, the main loop of QEMU is a translation-emulation loop. Before translating a basic block, QEMU checks whether it has been added to the code cache. If it can be found, the translated code in the code cache will directly be executed. We already know the entry basic block of the target loop by parsing related information files. Then, the code cache of the entry basic block is generated during initialization. Code fragments in the code cache are not the translated code from the basic block but a prologue to redirect the execution from the code cache to the parallelized code. The parameters are also transferred to indicate the base address and pointer of the state structure.

This method makes it possible to check for the entry of the target loops without performing a hash-table lookup when translating every basic block. The decision of triggering the switch of the control flow is hidden when finding the code cache, which QEMU originally needs to do. Nearly no runtime overhead is incurred by this method, except for the initialization of the target code cache.

3.5. Static Translator. Static translation starts from extracting loops from guest binaries. From the assembly code, the control flow graph (CFG) is reconstructed [10]. Code regions where the CFG cannot be reconstructed statically will be discarded. For there are always back edges in CFG of loops, it is easy to extract loops from guest binaries. Then, we attempt to extend these loops to the outermost to find loop nests, considering that we can benefit more from larger loop nests. All found loop nests are marked as a candidate for further optimization.

An initial estimate is made to check if we can benefit from parallelization of one loop nest. Here, we simply use loop counts as the threshold. We insert runtime check instructions to roll back to the bare sequential version if no benefits can be gained.

In the following, we show details about how we translate and parallelize these candidate loop nests.

3.5.1. Translating Machine Code to LLVM IR. The static translator translates the binary code of the target loops to the LLVM IR [9] for further optimization. To maintain the DBT emulation, as described in Section 3.3, the binary code is translated to TCG IR before conversion to LLVMIR. The translation is implemented by the TCG translator and the LLVM translator seamlessly.

The TCG translator is developed from the TCG of QEMU. We split the TCG from QEMU and extend it to form a static one. The TCG translator translates the machine code into the TCG IR at the granularity of a basic block. A basic block is defined as a single-entry single-exit region of a code with a control flow instruction at the end.

However, the original TCG cannot check whether the next instruction is the entry of another basic block and only finishes formulating a basic block when it meets a control flow instruction. As a result, basic blocks generated from loop nests by the TCG usually overlap, leading to complex control flows and redundant code.

To solve this problem, we insert a jump instruction to the succeeding instruction as the end of the basic block when we find that the next instruction is the entry of another basic block based on CFG.

Next, the TCG IR of basic blocks is taken by the LLVM translator as input. First, the TCG IR is seamlessly translated into LLVM IR by one-to-one mapping. Then, the basic blocks of one target loop are packaged as a function in LLVM IR, and attributes are added to provide prior information about DBT. Helper functions in TCG designed to emulate flags and soft-float computation are inlined. Thus, additional opportunities are provided to carry out loop-level optimization and make it easier for Polly to analyze these loop functions.

3.5.2. Optimization and Parallelization. The LLVM translator outputs an equivalent LLVM IR of loop nests with emulation instructions. However, the parallelizer cannot automatically parallelize target loops by directly taking lifted IR generated by the LLVM translator. In fact, as the parallelizer, Polly always fails by directly taking lifted IR as input. We also find that even for nested loops that can be parallelized by Polly in the form of a source code, IR translated from its sequential machine code cannot be optimized by Polly.

Meanwhile, the code organized at the loop level instead of the basic-block level provides us with opportunities to perform sophisticated optimization such as dataflow analysis. As QEMU performs only simple optimization within a basic block, there is still redundant code to be eliminated. Although there are many aggressive optimization passes, the LLVM optimizer cannot handle these issues completely owing to a lack of prior knowledge of the DBT emulation mechanism.

To overcome these issues, we present a feasible method for optimizing the unoptimized IR at the loop level and provide opportunities for polyhedral parallelism. The optimizer is developed on the basis of prior knowledge of DBT and implemented as LLVM passes to bridge the gap between the translator and Polly. Then, IR optimized by the optimizer will be taken into the parallelizer. Finally, the code with polyhedral optimization is output by Polly and compiled into an executable code as a shared library.

4. Towards Polyhedral Optimization

Since we try to adapt Polly which is designed for the source code to binaries, we use the static translator to transform binaries into LLVM IR. As described in Section 3.3, the TCG translator and the LLVM translator enable this transformation. However, directly taking the unoptimized IR

output by the LLVM translator as input, Polly cannot perform automatic parallelization on the binaries successfully. Therefore, to bridge the gap between the translated IR and the IR that Polly can handle, the optimizer has been inserted between the parallelizer and the LLVM translator in the static translator. Here, we present our method implemented in the optimizer for optimizing and transforming unoptimized IR into a Polly-friendly version.

According to our previous evaluation, we find that there are two major factors that impede Polly: (1) DBT emulation instructions are inserted into IR during translation from the machine code to TCG IR. As LLVM IR are converted from TCG IR by one-to-one mapping, these emulation instructions are not optimized. Complicated memory manipulation for DBT emulation makes IR complicated for Polly; (2) optimization during compilation makes the loop structure and symbolic information unavailable, and x86 registers are used for calculation and address computation. From these complex instructions, Polly cannot obtain the information it needs to model the target loop. In the following, we will elaborate on the solution to these two problems.

4.1. Loop-Level DBT-Specific Optimizations. In this subsection, we focus on redundant code elimination on translated IR. Only simple optimization like liveness analysis and store forwarding is carried out within basic blocks by the TCG translator. The LLVM translator transforms TCG IR to LLVM IR in a way of one-to-one mapping and integrates the basic block to the nested loop region with CFG completely known. Organized as loop nests instead of basic blocks, the code with high-level structure information makes it possible to carry out sophisticated optimization like data flow analysis and other aggressive optimizations.

LLVM passes such as common subexpression elimination and instruction combination can be used to optimize some rather simple redundant codes. But due to the lack of prior knowledge of the DBT mechanism, redundant memory manipulations related to the DBT emulation mechanism are overlooked. We all know that redundant memory access will incur considerable overhead.

In the following, we present our method to eliminate redundant memory manipulations caused by DBT emulation at the loop level.

4.1.1. Optimizing Virtual Register Emulation. The dynamic translator allocates the state structure in the global memory for each virtual processor to model its architecture state. Data in virtual registers are read and written by pointers into the state structure. When translating the x86 machine code into TCG IR at the granularity of a basic block, several load/store instructions are generated to alter the architecture state. However, nearly all such memory access to the state structure is redundant when the basic blocks are organized as a loop.

A typical example is given in Figure 3. We can see that the value in virtual register *%rax* is loaded immediately after the store instruction without any intervening stores to this state. Here, we can use the value *%rcx.0* to replace all use of the loaded value and thus forward the *store* to the *load*. After this

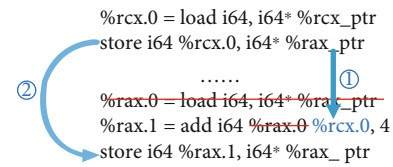


Figure 3: An example of substitution forwarding.

forward substitution, the store instructions are also exposed to be optimized by store sinking.

Then, we carry out such optimization at the loop level. With CFG completely known, such forward substitution and store sinking are performed across the loop nests. All redundant store and load instructions to update the virtual state structure within the loop bodies are eliminated. And only in the exit basic block, the store instructions are inserted to update the state structure according to the execution results of the last iteration.

4.1.2. Stack and Frame Manipulations. The stack and frame structure in x86 architecture is used to store the intermediate result of computation and program context. Memory accesses into stack and frame are performed by using frame pointer register *%rbp* and stack pointer register *%rsp*. As shown in Figure 4, the memory access into the stack is converted to a sequence of memory operations in LLVM IR. Such stack manipulation instructions cannot be handled by general loop passes.

If comparing stack memory cells in guest space to virtual states in QEMU space, we will find that memory manipulations on these memory cells behave the same. States in state structure can be addressed by pointers and constant offsets, while stack memory cells are addressed by *%rsp*/*%rbp* and offsets. To simplify complicated memory manipulations for emulating x86 stack and frame, we apply forward substitution and store sinking to these stack memory cells the same as to virtual states.

4.2. Tailoring IR for Polyhedral Optimization. Polly is a polyhedral optimization tool based on LLVM infrastructure. It is implemented as a set of LLVM passes, to perform analysis and transformations on IR. As a compiler tool, Polly is designed to optimize IR generated by Clang, and some passes in Polly rely on results of general pass such as the loop pass and SCEV pass.

Polly first formulates the polyhedral model of loops, starting with the detection of static control parts (SCoPs) [5] in loops. Only loop nest parts detected as SCoPs will be optimized by Polly. When a region cannot be detected as SCoPs, Polly will terminate the analysis on this region immediately.

Successful analysis of induction variables and memory access is essential to Polly. Polly obtains information about induction variables from the SCEV pass and uses the ISL library [11] to extract information about the stride and loop-trip count. A GEP instruction is a special instruction in LLVM IR that is used to calculate the target address of structural memory access. By taking the structure definition

x86 assembly	Unoptimized IR
L : … add rax , -0x8 (%rbp) … jl L …	L : … %rbp.0 = load i64, i64* %rbp_ptr %33 = add i64 %rbp.0, -8 %34 = inttoptr i64 %33 to i64* %35 = load i64, i64* %34 %36 = add i64 %rax.0, %35 store i64 36%, i64* %34 … br label L …

Figure 4: An example of stack manipulations.

of the array, pointer to array, index of target element, and type information as input, a GEP instruction returns the target address of array access. Polly obtains the structure information of array access from the GEP instructions.

From the above-mentioned description, we can determine that without heavy and architecture-specific optimization, loop structure information and symbolic information remain in IR generated by Clang. Passes on which Polly is dependent can obtain the loop structure information. However, translated IR is far more complicated than such unoptimized IR; hence, Polly always fails by directly taking the translated IR as input.

Next, we show these structure gaps and present our approach to seamlessly transform translated IR into a form suitable for Polly.

4.2.1. Memory Access Pattern. We already know that Polly obtains memory access patterns by analyzing GEP instructions, which involves the structure information of array access. However, address computation of array access in translated IR, by taking the values in virtual registers as operands, is performed similarly to that in the case of x86 architecture. As a result, Polly cannot reconstruct memory access patterns of translated IR.

Polyhedral optimization mainly targets affine loop nests, which requires the array index of array access of an affine function of induction variables. Symbolic descriptions are used to represent the array index of the affine memory address. For example, if i is the induction variable of array A, then $A[i]$ represents the ith element of array A and memory references $A[i]$ and $A[2i+1]$ are affine, whereas $A[i/4]$ is not.

To convert IR into a form suitable for Polly, it is crucial to know how the translated IR is organized to compute the address of array access in affine loops. On x86-64 architectures, the target address of memory access is usually computed as

$$\text{Target\_addr} = \text{Addr\_base} + s * \text{Index} + \text{Offset}, \quad (1)$$

where Addrbase and Index are values stored in registers, while s and Offset are immediate values. Such address computation can be performed within one instruction by the underlying x86 architectures.

There are structure gaps between LLVM IR and x86 assembly code. LLVM IR is of the SSA form, and every value is attached with explicit type information.

An example is shown in Figure 5. The x86 address computation operation is converted to multiple instructions in LLVM IR with the pattern of "calculation-inttoptr-bitcast." "Calculation" refers to operations that compute the target address by taking integers in virtual registers as input. Then, an inttoptr instruction is used to convert integer to pointer type in LLVM, taken by the load/store instruction as input. These structure gaps make it hard for Polly to model target loops.

4.2.2. Reconstructing Loop Structure. In the following, we present our method to reconstruct target loops from the translated IR. The key strategy is to recover the loop induction variables and memory access symbolic description. Based on recovered loop structure information, the translated IR is then transformed into an equivalent version with GEP instructions, which is suitable for polyhedral optimization on Polly.

To obtain symbolic description of memory access, we start from the PHI node in LLVM IR. With an initial value, the variable defined by the PHI node is modified in each iteration. We attempt to reconstruct the induction variable from these PHI nodes. The SCEV pass [12] is used to obtain the symbolic description of induction variables.

In particular, a normalized loop counter is introduced for each loop when there is no normalized one. Starting from zero, a normalized loop counter is incremented by one in every loop iteration. When the stride of a variable defined by the PHI node is constant, we can obtain its symbolic expression described by the normalized loop counter.

Then, we try to reconstruct the symbolic expression of memory access. By analyzing the pattern of calculation-inttoptr instructions, memory access is marked as a candidate for reconstruction. A symbolic expression of memory access is rebuilt using definitions involved in the address computation. This process is performed recursively by following def-use chains, until it reaches a known induction variable, an argument, or a loop-invariant variable.

When all the variables participating in the address computation in a loop, excluding loop-invariant variables, can

x86 assembly	Unpotimized IR
Addsd (%rcx, %rdx,1), %xmm0	%21 = add i64 %rcx, %rdx %22 = inttoptr i64 %21 to i64* %23 = bitcast i64* %22 to double* %24 = load double, double* %23

Figure 5: Difference of address computation.

be described by the affine functions of loop indices, this loop can be marked as a candidate.

For loop nests, the outer loop induction variable is usually the initial value of an inner loop induction variable. Such expression substitution and resolution are performed recursively from the innermost to the outermost one. Finally, we can always obtain the symbolic description of affine access in nested loops, expressed as

$$\text{Target\_addr} = \text{Addr\_base} + \sum_{k=1}^{n} \text{index}_k * \text{stride}_k, \quad (2)$$

where index_k is the induction variable of the kth loop dimension and stride_k is the loop stride. Addrbase is the starting address of an array.

When we obtain all the symbolic description of induction variables in nested loops, we have all the information of array accesses. However, to transform these instructions into GEPs, we need data type information of array elements. Note that in x86, variables in registers are not attached with explicit type information. Variables are stored in general-purpose registers. Type information of nearly all of them can be recovered only through the way it is used.

When translating an x86 instruction, TCG will specify a general type for a variable based on its size. When the general type is conflicting with operation input, TCG then convert it to the correct type. After computation, results are converted back to the original data type and wrote back to virtual registers or memory cells. Many *trunc* and *bitcast* instructions are inserted into IR to facilitate these data type conversions.

To solve this problem, we recognize the data type participating in calculation based on operations on data and unit size of the array element. To do this, we start from *inttoptr* and *bitcast* instructions to obtain correct data type information in calculation. Then, the correct data type is propagated by following def-use chains until we meet load instructions. As values are loaded from virtual registers or memory cells, we consider that we find the source of them. After data type resolution, the related instructions are modified to generate GEP instructions.

4.2.3. Separating Emulation Instructions. If a target loop is successfully reconstructed by our optimizer, Polly can detect SCoPs in it successfully and formulate the polyhedral representation. Then, dependence analysis is carried out to determine whether a loop is parallel. Even for loops which can be parallelized in the form of the source code, IR lifted from its machine code always fails to be marked as parallel by Polly. In this section, we attempt to address this issue.

We found that virtual register updating instructions introduce loop-carried dependencies to loop nests. As described in Subsection 4.1, expression forward substitution and store sinking are performed to eliminate redundant load/store instructions for emulating the virtual states. Only in the exit blocks of loop nests, virtual register states are updated according to intermediate results of instructions in the last iteration. However, for single-thread guest application, there is only one virtual CPU emulated by QEMU. These instructions with memory accesses to the virtual states introduce loop-carried dependencies to target loops because virtual states are global variables existing in QEMU space.

Since only intermediate results of instructions in the last iteration are used to alter virtual states, we separate the last iteration from the others to overcome this problem. Here, we assume that the loop-trip counts are constants, so it is easy to determine which is the last iteration. The target loop is split up into two regions: instructions in the last iteration and the kernel region with the other iterations. As a result, loop-carried dependencies caused by emulation instructions are removed from the kernel region. Then, the kernel region is fed to Polly to be parallelized.

5. Evaluation

5.1. Experimental Setup. In this section, we present the performance evaluation of LLPEMU. We conducted the experiments with the emulation of PolyBench4.2 benchmark [13], which consists of kernels in multimedia applications. Here, we choose 6 programs of them which are optimized well by Polly. We evaluated the performance of LLPEMU with x86_64-to-ARM emulation. With an ARM board as the host machine, LLPEMU takes x86_64 guest binaries (compiled by Clang -O2) as input and generates parallelized ARM binaries. The dynamic translator of LLPEMU is developed from QEMU version 5.1, and we use Polly version 6.0 as the polyhedral optimizer. The detailed hardware and software configurations are listed in Table 1.

5.2. Speedup of Loop-Level Optimizations. Firstly, we check the results of LLPEMU running test programs and confirm that the results generated by LLPEMU are the same as those from QEMU. The loops in kernels of test programs are successfully detected and replaced by the optimized code. Then, the performance of LLPEMU is compared to that of QEMU. We use the running time of QEMU translating kernel

Table 1: Hardware and software configurations.

Processor	ARMv8 (big.LITTLE)
Frequency	Up to 2.0 GHz
Cores	2 cortex-A72 cores+4 cortex-A53 cores
OS	Ubuntu 18.04
Linux kernel	4.4.179
Memory	2 GB

functions as the baseline. The dynamic running time of LLPEMU translating kernel functions is measured, and the speedup achieved by LLPEMU can be computed as

$$\text{Speedup} = \frac{\text{Time}_{\text{QEMU}}}{\text{Time}_{\text{LLPEMU}}}. \quad (3)$$

The performance of a simple version of LLPEMU only with loop-level optimizations and the full version with polyhedral optimization is shown in Figure 6. The y-axis is speedup ratios achieved by LLPEMU against QEMU. The results show that LLPEMU (simple) speeds up all six program executions and reaches an average of 5.0x. We also observe that LLPEMU (simple) achieves speedup ratios from 3.3x up to 9.0x.

The reason why loop-level optimizations can lead to such improvement mainly comes from two aspects. First, high-level structure information allows us to perform sophisticated optimizations, leading to high code quality. Second, taking the loop region as the translating unit reduces context switch overhead. When QEMU redirects its control to the execution of the translated code, the program context must be saved. And after execution, the context will be restored for control to come back to QEMU. These switches lead to lots of memory accesses which incur significant overhead.

Our second set of experiments is aimed at evaluating whether the method we proposed is feasible enough to transform unoptimized IR into the Polly-friendly version. Different from the first set of experiments, IR generated by the LLVM translator is optimized by the optimizer before taken into the parallelizer. Speedup times are calculated, also with the execution time of QEMU as the baseline.

From the results, we observe that the translated IR (except *syrk*) are parallelized by Polly successfully, and LLPEMU (full) gains considerable speedup times from the simple version with only loop-level optimization. Five of the benchmark programs achieve more than 6x speedup compared with QEMU, and four of them are more than 10x. The results demonstrate that our method successfully reconstructs loops in translated IR and transforms them into a form suitable for Polly. SCoPs in kernels have been detected and provide information to enable Polly to check whether they are parallel. It is clear that the method proposed to overcome obstacles preventing parallelization is feasible.

It is seen that LLPEMU (full) fails to perform parallelization transformation on *syrk* and achieves similar performance to its simple version. This comes from parametric loop bounds used in an inner loop of the *syrk* kernel. Our method cannot handle parametric loop bounds, and loops with it will not be marked as a candidate. Parametric loop bounds are usually stored in registers which require more complex analysis and symbolic derivation to recover loop bounds from binaries. Kotha et al. [14] have proposed a method to extract parametric loop bounds from registers in a pattern-matching way. Because Polly can analyze parametrized loops, we believe that this problem can be handled by LLPEMU integrated with more sophisticated analysis and transformations.

5.3. Quality of Parallelized Code. This evaluation is aimed at examining the quality of the parallelized code generated by LLPEMU and measuring how many opportunities our method provides for Polly to perform polyhedral optimizations. Figure 7 presents the performance of the native ARM code, kernel code generated by LLPEMU in 6 threads, and native ARM code parallelized by Polly. Here, the native code is compiled using "Clang -O2." "Polly-6" represents for the native code parallelized by Polly in 6 threads.

From the results, we observe that the kernel code generated by LLPEMU is even better than the native code with an average 3.09x speedup. This observation supports the importance of leveraging polyhedral optimization in DBT. Because kernels in benchmark programs are all loop nests with simple computation, parallelization results in a large portion of performance improvement. Thus, the performance of code generated by LLPEMU differs from that of "Polly-6" slightly.

Next, we move to the comparison of LLPEMU and "Polly-6." From the results, we observe that LLPEMU on *gemver* is only about 40% of "Polly-6." The reason that causes slowdown is that our method fails to eliminate all loop-carried dependencies. We found that loop-level optimizations eliminate most of those dependencies, but it is still required to perform more advanced optimizations to eliminate them all. If we eliminate such dependencies, we could still achieve performance improvement like the other four programs.

It is also seen that LLPEMU is always slower than "Polly-6" with an average of 90% excluding *gemver*. The reason for this phenomenon is that there are many additional instructions in kernels generated by LLPEMU. As described previously, assumptions are made and runtime validation needs to be done in kernels to make sure those assumptions are right. If not, kernels will roll back to a sequential version instead of a parallelized one. Besides, emulation instructions and manipulations to package the kernel function to a Polly-friendly version also reduce the speed.

6. Related Work

The novelty of this study is the proposal of a feasible way to improve the performance of a dynamic binary translation system by leveraging polyhedral optimization at the loop level. Beyond automatic parallelization, this study provides opportunities for further optimization, such as automatic vectorization [15] and hotspot offloading [16], to platforms with more powerful computation resources.

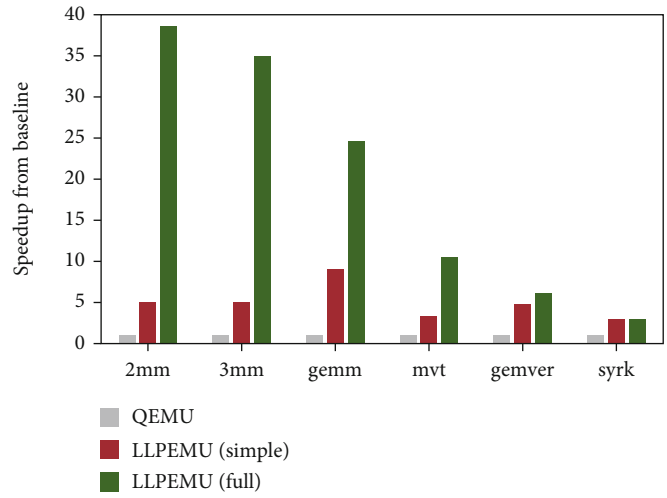


Figure 6: Performance comparisons of LLPEMU and QEMU.

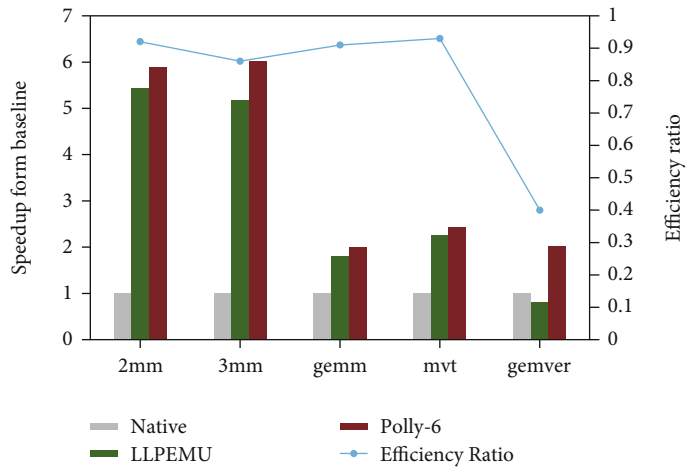


Figure 7: Quality of parallelized code.

By leveraging DBT for various purposes, such as cross-ISA emulation, transparent optimization [17], and profiling [18], many optimization approaches have been proposed. This paper is aimed at achieving better performance when migrating multimedia applications across ISAs using DBT. To this end, we leverage polyhedral optimization to generate the concurrent code that exploits abundant multicore resources.

6.1. Binary Translation. To the best of our knowledge, the approach that is the most closely related to LLPEMU is HQEMU [7]. HQEMU is a trace-based dynamic binary translator that is also built on QEMU and uses LLVM as the backend optimizer. Small sections of the code are inserted at the beginning of each translated code region to obtain profiling information. Based on profiling information, trace detection and merging are performed. Then, the traces are optimized and translated into the host machine code by optimizers on different processor cores. This approach involves profiling and trace optimizing at runtime, and it does not carry out polyhedral optimization to generate the concurrent code. In its DBT system, optimizers are attached to different processors to reduce the runtime overhead. By contrast, LLPEMU detects and parallelizes loops statically. Thus, with high-level information of the target loops, sophisticated optimization can be performed while incurring low runtime overhead. Furthermore, instead of using multiple cores to reduce the optimization overhead, LLPEMU generates a concurrent code to directly improve the execution performance.

Some other multithread DBT systems have also been proposed. COREMU [19] emulates multiple cores in the full-system model. Its system is parallelized by creating multiple QEMU emulators and assigning them to multiple threads. In a different way, Ding and Chang [20] parallelize QEMU internally and propose a method to arrange critical sections carefully. Their goal is to improve the performance of emulating multithread applications by parallelizing DBT systems, and they did not perform sophisticated optimization on the translated code. Their DBT systems cannot benefit as much when translating sequential applications, whereas our system

is mainly aimed at parallelizing loops in sequential applications.

6.2. Binary Parallelization. There has been some prior work in binary parallelization. Sato et al. [21] proposed a dynamic code parallelization system (ExanaDBT) that also applies Polly to LLVM IR recovered from the binary code. However, their approach cannot work with DBT, and they assume that target loops do not contain access to global variables. By contrast, virtual states in binary translation are stored as global variables, and most of our work is related to these global variables. Pradelle et al. [22] partly lift binaries to C code and fed them to a polyhedral compiler. Kotha et al. [14] proposed a static binary parallelizer, which uses a dependence vector to determine whether a given loop should be parallelized. However, all these approaches are aimed at transparently parallelizing binaries within the same ISA. Instead, our system translates and parallelizes loops in guest binaries into a cross-ISA concurrent code and supports various architectures, as it uses retargetable tools QEMU and LLVM.

7. Conclusions and Future Work

In this paper, we presented LLPEMU, a dynamic binary translation framework that automatically parallelizes the guest binary code. LLPEMU translates and parallelizes loop nests in the guest binary code statically and switches to the parallelized code at runtime. The static-dynamic combined design allows LLPEMU to perform analysis and polyhedral optimization with low runtime overhead. Further, we investigated factors that impede parallelization of translated IR by the polyhedral optimizer of LLPEMU and proposed a feasible method to overcome these obstacles.

We have evaluated the performance of LLPEMU on the PolyBench benchmark. The results show that LLPEMU successfully performed translation and parallelization on loop nests in x86_64 binaries and achieved a considerable performance improvement over QEMU and even over the native sequential code in some cases.

In the future, this work must be extended and improved from the following aspects: (1) performing more powerful analysis and more aggressive optimizations to remove data dependencies in reconstructed loop nests and (2) extending the ability of our method to handle binaries compiled with more aggressive optimizations.

Data Availability

All the data is available online.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China under Grant 61472447, Grant 61802433, and Grant 61802435.

References

[1] E. R. Altman, D. Kaeli, and Y. Sheffer, "Welcome to the opportunities of binary translation," *Computer*, vol. 33, no. 3, pp. 40–45, 2000.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[3] J. Shuja, A. Gani, M. H. Rehman et al., "Towards native code offloading based MCC frameworks for multimedia applications: a survey," *Journal of Network and Computer Applications*, vol. 75, pp. 335–354, 2016.

[4] F. Bellard, "QEMU, a fast and portable dynamic translator," in *USENIX Annual Technical Conference, FREENIX Track*, vol. 41, p. 46, Berkeley, CA, USA, 2005.

[5] T. Grosser, A. Groesslinger, and C. Lengauer, "Polly—performing polyhedral optimizations on a low-level intermediate representation," *Parallel Processing Letters*, vol. 22, no. 4, article 1250010, 2012.

[6] Tiny code generatorhttp://wiki.qemu.org/Documentation/TCG.

[7] D. Y. Hong, C. Hsu, P. Yew et al., "HQEMU: a multi-threaded and retargetable dynamic binary translator on multicores," in *Proceedings of the Tenth International Symposium on Code Generation and Optimization*, pp. 104–113, San Jose, CA, USA, 2012.

[8] A. Mignone, G. Bodo, S. Massaglia et al., "PLUTO: a numerical code for computational astrophysics," *The Astrophysical Journal Supplement Series*, vol. 170, no. 1, pp. 228–242, 2007.

[9] C. Lattner and V. Adve, "LLVM: a compilation framework for lifelong program analysis & transformation," in *International Symposium on Code Generation and Optimization, CGO 2004*, pp. 75–86, Palo Alto, CA, USA, 2004.

[10] Y. Sato, Y. Inoguchi, and T. Nakamura, "Identifying program loop nesting structures during execution of machine code," *IEICE Transactions on Information and Systems*, vol. E97.D, no. 9, pp. 2371–2385, 2014.

[11] S. Verdoolaege, "isl: an integer set library for the polyhedral model," in *International Congress on Mathematical Software*, pp. 299–302, Springer, Berlin, Heidelberg, 2010.

[12] R. V. Engelen, "Efficient symbolic analysis for optimizing compilers," in *International Conference on Compiler Construction, (CC2001)*, pp. 118–132, Springer, Berlin, Heidelberg, 2001.

[13] PolyBench PolyBench, 2017, https://sourceforge.net/projects/polybench/.

[14] A. Kotha, K. Anand, M. Smithson, G. Yellareddy, and R. Barua, "Automatic parallelization in a binary rewriter," in *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 547–557, Atlanta, GA, USA, December 2010.

[15] N. Hallou, E. Rohou, and P. Clauss, "Runtime vectorization transformations of binary code," *International Journal of Parallel Programming*, vol. 45, no. 6, pp. 1536–1565, 2017.

[16] M. Damschen, H. Heinrich, G. Vaz, and C. Plessl, "Transparent offloading of computational hotspots from binary code to Xeon Phi," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*, pp. 1078–1083, Grenoble, France, 2015.

[17] M. A. Watkins, T. Nowatzki, and A. Carno, "Software transparent dynamic binary translation for coarse-grain

reconfigurable architectures," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 138–150, Barcelona, Spain, March 2016.

[18] I. Böhm, B. Franke, and N. Topham, "Cycle-accurate performance modelling in an ultra-fast just-in-time dynamic binary translation instruction set simulator," in *2010 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, pp. 1–10, Samos, Greece, July 2010.

[19] Z. Wang, R. Liu, Y. Chen et al., "COREMU: a scalable and portable parallel full-system emulator," in *Proceedings of the 16th ACM symposium on Principles and practice of parallel programming - PPoPP '11*, pp. 213–222, San Antonio, TX, USA, 2011.

[20] J. H. Ding and P. C. Chang, "PQEMU: a parallel system emulator based on QEMU," in *Proceedings of the 2011 IEEE 17th International Conference on Parallel and Distributed Systems, IEEE Computer Society*, vol. 10, Washington, 2011.

[21] Y. Sato, T. Yuki, and T. Endo, "ExanaDBT: a dynamic compilation system for transparent polyhedral optimizations at runtime," in *Proceedings of the Computing Frontiers Conference*, pp. 191–200, Siena, Italy, May 2017.

[22] B. Pradelle, A. Ketterlin, and P. Clauss, "Polyhedral parallelization of binary code," *ACM Transactions on Architecture and Code Optimization*, vol. 8, no. 4, pp. 1–21, 2012.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6656564, 11 pages
https://doi.org/10.1155/2021/6656564

Research Article

Convolutional Neural Network for Voltage Sag Source Azimuth Recognition in Electrical Internet of Things

Ding Kai,[1] **Li Wei**,[1] **Sun Jianfeng**,[2] **Xiao Xianyong**,[2] **and Wang Ying**[2]

[1]*Hubei Electric Power Research Institute State Grid, 430077 Wuhan, China*
[2]*College of Electrical Engineering, Sichuan University, 610065 Chengdu, China*

Correspondence should be addressed to Ding Kai; shellding@163.com

Received 23 October 2020; Revised 3 December 2020; Accepted 20 February 2021; Published 22 March 2021

Academic Editor: Shaohua Wan

Recognition and analytics at the edge enable utility companies to predict and prevent problems in real time. Clearing the voltage sag disturbance source by the positioning method is the most effective way to solve and improve the voltage sag. However, for different grid structures and fault types, the existing methods usually achieve a sag source location based on the single feature of monitoring data extraction. However, due to the effectiveness and applicability of the existing method features, this paper proposes a multidimensional feature of the voltage sag source positioning method of the matrix. Based on the analysis of the characteristics of the voltage sag event caused by the fault, this paper proposes a multidimensional feature matrix for voltage sag source location, based on the convolutional neural network to establish the mapping relationship between the feature matrix and the voltage sag position, thus achieving multiple points based on multiple points. The voltage sag source orientation is identified by the monitoring data. Finally, the voltage sag event caused by the short-circuit fault is simulated in the IEEE14 node model, and the effectiveness of the proposed method is verified by simulation data. The simulation results show that the proposed method has higher accuracy than the traditional method, and the method can be applied to different grid structures and different types of faults.

1. Introduction

Electric Internet of Things (UEIoT) realizes the interconnection of everything and human-computer interaction in all aspects of the power system by fully applying mobile Internet technology, artificial intelligence technology, and advanced communication technology. It provides intelligent services characterized by a comprehensive perception of the state, efficient processing of information, and convenient and flexible application. In order to drive reliability and operational efficiency, utility companies now need to leverage advanced distribution analysis. Delivering analytics at the edge means that the energy industry can use data in a far more effective and efficient way. Detection at the edge enables utility companies to predict and prevent problems in real time, to cost-effectively deploy resources and personnel, and to increase overall grid optimization, security, and reliability. A voltage sag is a power quality disturbance event where supply voltage quickly recovers after dropping [1]. The root cause of the voltage sag is the voltage sag of a common connecting point which is caused by the increase in the partial voltage of power impedance due to the increase in current. The causes of the voltage sag can be divided into short-circuit fault, transformer magnetizing inrush current, inductive motor start, and other large-capacity load operations [2–4]. The positioning of the voltage sag can not only be used to divide the temporary responsibility of both power supply and power consumption but also help to shorten the time spent by a power supply company on fault elimination [5].

At present, the positioning methods of the current voltage sag can be roughly divided into positioning methods based on a single monitoring point [6–9] and positioning methods based on multiple monitoring points [10, 11]. Literature [6] determines whether the voltage sag source is in the upstream or downstream of the power quality detection device according to the signs of the disturbed active power (i.e., the difference between the active power before and after

the occurrence of the voltage sag) and the disturbed active power energy (the integral of the disturbed active power on a time scale). Literature [7] uses the fundamental voltage and current recorded by the power quality monitoring device to calculate equivalent impedance and determines whether the voltage sag is located in the upstream or downstream of the power quality monitoring device according to the signs of the real part of impedance. Literature [8] judges that the voltage sag source is located in the upstream or downstream of the monitoring point according to the slope of the line segment connected by two points before and after the fault. Literature [9] determines whether the voltage sag source is located in the upstream or downstream of the monitoring point according to the signs of the real part of current during the voltage sag. Literature [10] puts forward the sensitivity index of the fault current to the voltage sag, determines the priority of nodes according to the sensitivity index, and then determines the priority of the installation location of the power quality monitor according to this index. Then, the position of the voltage sag source is determined according to the current variation of each monitoring branch before and after the voltage sag event as well as the priority of the nodes. Literature [11] searches the voltage sag path by using the feeder sending end and determines that the branch line with the lowest voltage is the line where the voltage sag occurs. Then, a quadratic function is used to fit the voltage into the function of distance, which can not only realize the estimation of node voltage sag but also achieve the location of the voltage sag source.

Traditional locating methods of the voltage sag source have some limitations in practical application. The locating methods based on a single monitoring point are only suitable for a radiation network. Besides, the judgment results based on the disturbed active power and disturbed active energy may not match [2], and this method lacks theoretical derivation [12]. The real part method of equivalent impedance [7], the system trajectory slope method [8], and the real part current method [9] all assume that the circuit conditions will not change when there is a voltage sag caused by the fault, which is bound to affect the judgment result. Although the method proposed in literature [10] is applicable to radiation networks and ring networks, it is affected by the system grounding mode. The method proposed in literature [11] is only applicable to radial networks. To solve the problems such as limited application scope and inaccurate positioning of the existing methods, literature [3, 13, 14] integrates the system slope trajectory and other traditional positioning methods as comprehensive criteria and uses the BP Neural Network (BPNN) and Support Vector Machine (SVM) to establish the mapping relationship between the criteria and positioning results, respectively. Literature [15, 16] obtains the first peak of the disturbed power after the occurrence of the voltage sag as the feature vector, and then, the SVM and Radial Basis Function (RBF) Neural Network are used to conduct classification and then achieve fault location, respectively. The voltage sag source location method based on classification considers the voltage sag source location a binary classification problem and determines whether the voltage sag source is located in the upstream or downstream of a single monitoring point. Such methods are prone to have conflicting features that affect the classification results, especially when they are used for voltage sag positioning in a loop network.

As the power grid fault is the main cause of the voltage sag [17, 18], in this paper, on the basis of the research of the existing methods, the main contributions are as follows:

(1) A voltage sag source location method based on a multidimensional feature matrix is proposed to conduct the positioning of the voltage sag events caused by the power grid fault

(2) Then, the characteristics of voltage sag events are analyzed, and the disturbance power, disturbance energy, real part of equivalent impedance, slope trajectory, and real part of the current of all monitoring nodes of the whole network are extracted to form the feature matrix

(3) Later, the locating problem of the voltage sag source is transformed into a multiclassification problem. The features are extracted by the convolutional layer and sampling layer of the convolutional neural networks (CNN) to avoid the contradiction between the features. The mapping relationship between deep features and the location of the voltage sag source is established through the fully connected layer of CNN to realize the location of the voltage sag source

(4) In the IEEE14 node model, the grid fault data is obtained by simulation, in order to verify the effectiveness of the proposed method. The simulation results show that the proposed method is more accurate than the traditional methods and more applicable to different network structures and different types of faults

The rest of the paper is organized as follows. We first discuss some related works in voltage sag source location methods, and then, we elaborate our proposed sag source location method based on convolutional neural network algorithm. Moreover, we present experiments to evaluate our method in Simulation Analysis. Finally, the conclusions of this paper are given in Conclusion.

2. Voltage Sag Source Location Methods

2.1. Upstream and Downstream Positioning Methods. Both positioning criteria [6–9] proposed according to the variation of electric quantity when the voltage sag occurs and the method that constitutes a comprehensive criterion for a classifier's classification and identification [13–16] judge whether the voltage sag source is located in the upstream or downstream of the power quality monitoring device. As shown in Figure 1, the reference direction of the power flow has been marked with arrows. As for the power quality monitoring device M, the lines L_1 and L_4 where the energy flows to M and the lines L_2, L_3, and L_{10} where the energy does not flow through M are the upstream regions, while the lines $L_5 \sim L_9$

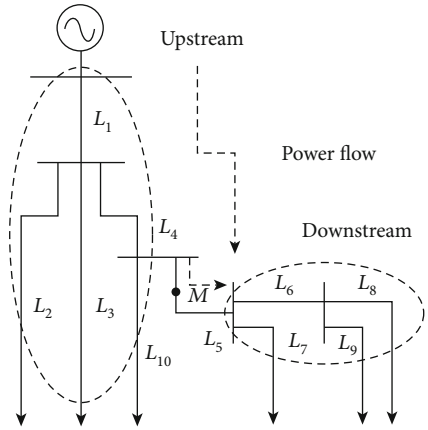


Figure 1: Voltage sag source location.

where the energy flows through M are the downstream regions. This method which defines upstream and downstream is only applicable to radial networks, but for more complex networks such as ring networks, it may be difficult for this method to define upstream and downstream.

2.2. Multimonitoring Point Positioning Method. The positioning method based on multimonitoring points needs to install a large number of monitoring devices in the power grid. At present, the positioning methods based on multimonitoring points mainly include the deviation method based on branch current [10] and the node voltage deviation index method [11].

The branch current deviation method uses the system coefficient matrix to calculate the current of the system when the bus k fails:

$$f\left(V_f^k\right) = \sum_{\substack{i=1\\ i\neq j}}^{m} \sum_{\substack{j=1\\ j\neq i}}^{m} I_{ij}^k = \sum_{\substack{i=1\\ i\neq j}}^{m} \sum_{\substack{j=1\\ j\neq i}}^{m} \frac{\left(z_{ik} - z_{jk}\right)\left(V_f^k / \left(z_{kk} + z_f^k\right)\right)}{z_{b,ij}}, \tag{1}$$

where z_{kk} is the equivalent resistance at the bus k, z_f^k represents the fault resistance, z_{ik} is the transmission impedance between the buses i and k, and $z_{b,ij}$ represents the line impedance between the buses i and j.

The sensitivity of the system current to the transient voltage is defined as

$$\frac{\partial f}{\partial V_f^k} = \sum_{\substack{i=1\\ i\neq j}}^{m} \sum_{\substack{j=1\\ j\neq i}}^{m} \frac{\left(z_{ik} - z_{jk}\right)}{z_{b,ij}\left(z_{kk} + z_f^k\right)}. \tag{2}$$

The branch current deviation is

$$\mathrm{BCD} = \frac{I_{\mathrm{sag}} - I_{\mathrm{pre}}}{I_{\mathrm{pre}}}, \tag{3}$$

where I_{pre} and I_{sag} are defined as the current values before and after the failure, respectively.

The sensitivity of all nodes in the network is calculated, and the monitoring device is installed in the nodes with higher sensitivity. When a fault occurs, the current deviation of each branch of the bus equipped with the monitoring device is calculated, and the positioning of the voltage sag source can be realized by searching according to the current deviation. This method has obvious defects, and the fault current varies in different grounding modes, so it is difficult to guarantee the reliability of its application.

The node voltage deviation method calculates the voltages of all nodes in the whole network based on the system coefficient and monitoring data. The voltage deviation is defined as

$$V = \frac{V_{\mathrm{pre}} - V_{\mathrm{sag}}}{V_{\mathrm{pre}}} \times 100\%. \tag{4}$$

The bus $V_{\max} = [V_1, V_2, V_3 \cdots V_n]$ with the maximum voltage deviation is determined as the location where the voltage sag source occurs. This method does not take into account the voltage sag caused by different short-circuit faults, so it has some limitations.

3. Multidimensional Characteristic Matrix for Sag Source Location

Effective feature selection is the key to realize classification. Based on the existing methods, the disturbance power and energy, the real part of the equivalent impedance, the slope of the system trajectory, and the real part of the current are extracted from each node.

3.1. Disturbance Power and Disturbance Energy. DP (disturbance power) is defined as the difference between the instantaneous power and the steady-state operating power, while DE (disturbance energy) is defined as the integral of the disturbance energy during the disturbance.

$$\mathrm{DP} = p(t)_{\mathrm{f}} - p(t)_{\mathrm{s}}, \tag{5}$$

$$\mathrm{DE} = \int_0^t \mathrm{DP}(t)dt, \tag{6}$$

where $p(t)_{\mathrm{f}}$ and $p(t)_{\mathrm{s}}$ represent the transient power during disturbance and steady-state operation, respectively.

3.2. Equivalent Impedance Real Part. Based on the equivalent circuit shown in Figure 2, the equivalent impedance variation at the power quality monitoring device caused by the voltage sag is analyzed.

Assuming that there is a disturbance in the downstream of the power quality monitoring device, the system

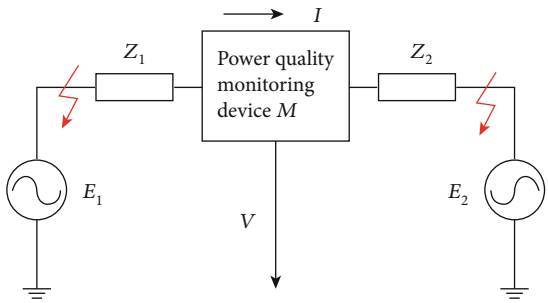


Figure 2: Equivalent circuit of the voltage sag.

parameters do not change during the disturbance, the voltage variation is ΔV, and the current variation is ΔI, then

$$V = E_1 - IZ_1, \tag{7}$$

$$V + \Delta V = E_1 - (I + \Delta I)Z_1. \tag{8}$$

The equivalent impedance can be calculated by Equations (7) and (8):

$$Z_2 = \frac{\Delta V}{\Delta I}. \tag{9}$$

Similarly, the equivalent impedance can be calculated when the disturbance occurs in the upstream:

$$Z_1 = -\left(\frac{\Delta V}{\Delta I}\right). \tag{10}$$

Therefore, the equivalent impedance can be defined as $Ze = \Delta V/\Delta I$, and the voltage sag source can be localized according to the polarity.

3.3. System Slope Trajectory.

$$VI\cos\theta_2 = -I^2\,\mathrm{Re}\,(Z_1) + E_1 I\cos\theta_1. \tag{11}$$

Equation (10) can be transformed into

$$V\cos\theta_2 = -I\,\mathrm{Re}\,(Z_1) + E_1\cos\theta_1, \tag{12}$$

where θ_2 denotes the phase difference between V and I and θ_1 represents the phase difference between E and I.

When the disturbance occurs in the upstream of the power quality monitoring device, $\cos\theta_2 < 0$; when the disturbance occurs in the downstream of the power quality monitoring device, $\cos\theta_2 > 0$. It can be seen from Equation (11) that, assuming that the operating parameters do not change when the disturbance occurs, the position of the voltage sag can be judged by the correlation between $V\cos\theta_2$ and I. Therefore, a line is synthesized during the disturbance to locate the voltage sag source according to the slope of the line k.

3.4. Real Part of the Current. According to Equation (12), when the disturbance occurs in the upstream, the current direction during the disturbance is opposite to that before the disturbance. When the disturbance occurs in the downstream, the current direction during the disturbance is the same as that before the disturbance. Therefore, the sag source can be localized according to the polarity.

3.5. Characteristic Matrix. The characteristic matrix of the whole network is as follows:

$$M_i = \begin{bmatrix} \mathrm{DP}_1 & \mathrm{DP}_2 & \cdots & \mathrm{DP}_n \\ \mathrm{DE}_1 & \mathrm{DE}_2 & \cdots & \mathrm{DE}_n \\ Ze_1 & Ze_2 & \cdots & Ze_n \\ k_1 & k_2 & \cdots & k_n \\ I\cos\theta_1 & I\cos\theta_2 & \cdots & I\cos\theta_n \end{bmatrix}, \tag{13}$$

where $M_i (i \in \{a, b, c\})$ represents the characteristic matrix of phase i, and the three-dimensional characteristic matrix formed is used as the input of the three channels of CNN.

4. Sag Source Location Method Based on the Convolutional Neural Network (CNN)

The voltage sag location method proposed in this paper is used to classify and identify voltage sag events occurring on different lines. When the 3D characteristic matrix selected in Section 1 is used for classification and recognition, there may be a problem of mutual redundancy among features or failure to completely represent the voltage sag, which affects the accuracy of the final classification. As a bionic model, CNN's basic framework usually includes a feature extractor and classifier [19], and it is composed of a convolutional layer, sampling layer, and fully connected layer [20–22]. Its structure is shown in Figure 3. The convolutional layer and the sampling layer can not only effectively learn the original feature matrix of the input at a deeper level but also reduce the number of neurons and simplify the complexity of the network through weight sharing. The fully connected layer, as a classifier, inputs the features after convolution and sampling and outputs classification categories.

4.1. Forward Pass. In the forward pass process, the output of the lth convolutional layer is

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j\right), \tag{14}$$

where x_j^l denotes the jth output characteristic matrix of the lth layer, f denotes the activation function, M_j denotes the combination of the characteristic matrix output by the upper layer, k_{ij}^l denotes the convolution kernel connecting x_j^l and x_i^{l-1}, and b_j denotes the bias corresponded by x_j^l.

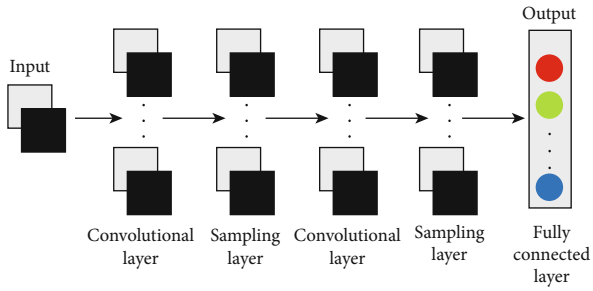


Figure 3: The structure of CNN.

The output of the lth convolutional layer is the input of the lth sampling layer. In this case, the output of the lth sampling layer is

$$x_j^l = \text{down}\left(x_j^{l-1}\right) + b_j^l, \tag{15}$$

where down() represents the lower sampling function.

The output obtained by convolution sampling is sorted into $X = \{x_1, x_2, \cdots, x_n\} \in \mathbb{R}^n$, as the input of the fully connected layer, and the output of the lth fully connected layer is

$$X^l = f\left(W_{l-1}^l X^{l-1} + b^l\right), \tag{16}$$

where W_{l-1}^l denotes the weight connecting X^{l-1} and X^l and b^l is the bias of the lth layer.

4.2. Backward Pass. The training of CNN minimizes the error between the output data and the expected data by adjusting the weight and bias, and this process is known as reverse pass. As a supervised learning algorithm, it is necessary to define a loss function before conducting reverse pass. The commonly used error functions include the square error loss function and cross-entropy loss function. In this paper, the cross-entropy loss function is selected:

$$\text{Loss}(W, b, x, y\_) = -\frac{1}{n}\sum[y\_ \cdot \ln y + (1 - y\_) \cdot \ln (1 - y)], \tag{17}$$

where n represents the number of samples, $y\_$ represents the expected output, and y represents the actual output.

According to the loss function, the error of each neuron in the lth output layer can be calculated:

$$\delta_j^l = \frac{\partial L}{\partial X_j^L} f'\left(z_j^l\right), \tag{18}$$

where z_j^l is the value of the input activation function of the jth neuron in the lth layer. Equation (17) can be written as the matrix form as follows:

$$\delta^l = \nabla_X L \odot f'\left(Z^l\right), \tag{19}$$

where $\odot$ represents the Hadamard product.

The error of the lth layer of the fully connected layer can be used to calculate the error of the $l-1$th layer of the full connected layer:

$$\delta^{l-1} = \left(\left(W^l\right)^T \delta^l\right) \odot f'\left(Z^{l-1}\right). \tag{20}$$

Given the sampling layer error, the error of the previous layer can be calculated as follows:

$$\delta^{l-1} = \text{upsample}\left(\delta^l\right) \odot f'\left(Z^{l-1}\right), \tag{21}$$

where upsample() represents the upsampling function.

Given the convolution layer error, the error of the previous layer can be calculated as follows:

$$\delta^{l-1} = \delta^l * \text{rot180}\left(W^l\right) \odot f'\left(Z^{l-1}\right). \tag{22}$$

The backpropagation of the error can be used to calculate the gradient of the weight and bias:

$$\begin{aligned} \frac{\partial L}{\partial w_{jk}^l} &= Z_k^{l-1}\delta_j^l, \\ \frac{L}{b_j^l} &= \delta_j^l. \end{aligned} \tag{23}$$

The gradient descent method is adopted to adjust the weight and bias of the network:

$$Wb = Wb - \beta\frac{\partial L}{\partial Wb}, \tag{24}$$

where Wb denotes the weight or bias and β denotes the learning rate.

4.3. Sag Source Positioning Method. In this paper, a voltage sag source positioning method based on a multidimensional characteristic matrix is proposed to locate single voltage sag events caused by the grid fault. The positioning of the voltage sag source is considered a multiclassification problem, and the voltage sag events occurring on different lines are divided into different categories. The mapping relationship between voltage sag events and lines is established by CNN to realize the location of the voltage sag source. The flowchart of the method proposed in this paper is shown in Figure 4.

(1) Record the voltage and current waveform of all monitoring points in the whole network when the voltage sag occurs. The IEEE14 node model is built in PSCAD/EMTDC to simulate voltage sag events caused by the short-circuit fault, and then, the voltage waveform of each node corresponded by each voltage sag event and the current waveform at both ends of each line are recorded

(2) Based on the data recorded in step (1), the disturbance power, disturbance energy, real part of

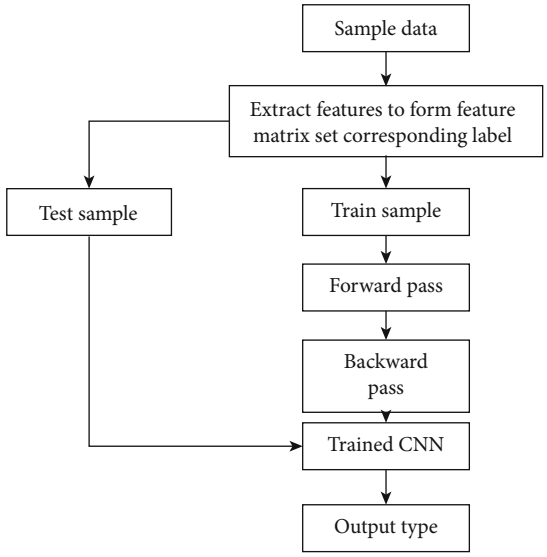


Figure 4: Flowchart of the method.

equivalent impedance, slope of system trajectory, and real part of the current at each end of each line are calculated to form a feature matrix, and then, the corresponding labels are set according to the line where the voltage sag occurs. The characteristic matrix and corresponding tags are used as a sample for CNN training. In this paper, the CNN trained for the test randomly selects a part of the obtained samples as the test samples

(3) Take the training sample as the input of CNN and conduct forward pass to output the predicted tag. In order to make CNN output the result that matches the actual tag, reverse pass is required to adjust the network parameters. The trained CNN has established the mapping relationship between input and output

(4) Based on the CNN completed by training, the positioning of the voltage sag source can be realized. Input the characteristic matrix of voltage sag events that need to be located, and the output is the label of the circuit that occurs. This paper tests the classification performance of CNN completed by training based on the test samples obtained in step (2).

5. Simulation Analysis

5.1. Data Acquisition. To verify the CNN-based voltage sag source location method proposed in this paper, a large amount of data of voltage sag events are needed. Firstly, the IEEE14 node model is built in PSCAD/EMTDC, as shown in Figure 5. Faults are set at each ten equal points of each line, including single line-to-ground fault (SLGF), phase-to-phase fault (PPF), double line-to-ground fault (DLGF), and three line-to-ground fault (TLGF). Ten uniform random numbers whose transition resistance is between 0 and 10 Ω are set at each fault point for simulation. Then, the sampling frequency of 5 kHz is used to record the voltage waveform of 14 nodes

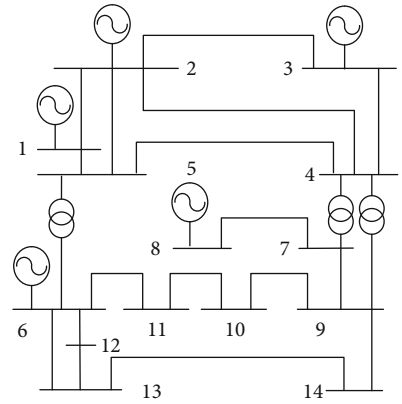


Figure 5: IEEE14 model.

Table 1: Distribution of samples.

Type	Sample size	Training sets	Testing sets
SLGF	1700	1360	340
PPF	1700	1360	340
DLGF	1700	1360	340
TLGF	1700	1360	340

and the current waveform at both ends of 17 lines during each simulation.

The recorded waveform data is input into MATLAB for processing. First of all, an FIR low-pass filter is designed to filter the collected data and retain the power frequency component. Secondly, the features mentioned in Section 2 at both ends of each line are extracted to form a feature matrix of $5 \times 34 \times 3$, and then, the corresponding label (label format is $5 \times 34 \times 3$) is set. The 100 sample sets of each line are randomly divided into 80 training samples and 20 test samples, and the sample distribution is shown in Table 1.

5.2. Model Evaluation Index. The indexes which are used to evaluate dichotomy problems mainly include accuracy rate, recall rate, F1 value index, ROC curve, etc. [23–26], but they are no longer fully applicable to the multiple classification problems appearing in this paper. Therefore, the accuracy rate and kappa coefficient [27–30] are intended to be used to evaluate the models used in this paper.

The calculation formula of kappa is as follows:

$$K = \frac{P_0 - P_e}{1 - P_e}, \tag{25}$$

$$P_e = \frac{a_1 b_1 + a_2 b_2 + \cdots + a_i b_i}{N \cdot N}, \tag{26}$$

where P_0 represents the classification accuracy of samples, a_i represents the number of samples of category, b_i represents the number of samples predicted to be category i, and N represents the total number of samples.

5.3. Construction of the CNN Model. Based on TensorFlow, a deep learning framework developed by Google, CNN is built.

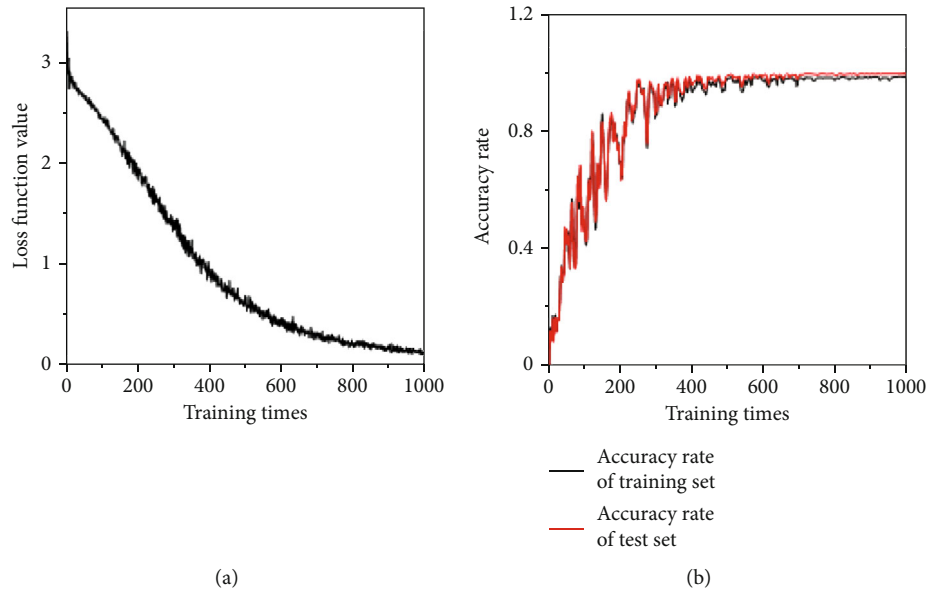


Figure 6: Training process: (a) loss function value curve and (b) accuracy rate curve.

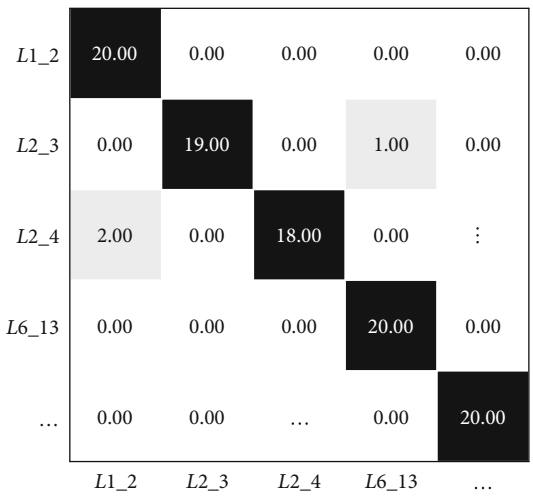


Figure 7: Confusion matrix of classification results.

Table 2: Classification indicator.

Classification algorithm	Classification feature	Accuracy	Kappa
CNN	Characteristic matrix	99.12%	99.06%
KNN	Characteristic matrix	97.06%	96.88%
	Disturbance power	43.53%	40.00%
	Equivalent impedance	54.41%	51.56%
	System slope	51.76%	48.75%
	Real part of current	57.94%	55.31%
EL	Characteristic matrix	87.94%	87.19%
	Disturbance power	40.29%	36.56%
	Equivalent impedance	57.06%	54.38%
	System slope	41.18%	37.50%
	Real part of current	32.94%	28.75%

The CNN consists of two convolutional sampling layers and three fully connected layers [31–36]. The first convolutional layer has 32 convolution kernels with a size of 2×5, the second convolutional layer has 64 convolution kernels with a size of 1×6, the convolution kernel of the sampling layer is 2×2, and the number of nodes in the middle layer of the fully connected layer is 1024. Among them, the activation function of the hidden layer is "RELU," the activation function of the output layer is "SOFTMAX," the convolution mode is "VALID," and the sampling mode is "SAME."

The CNN which has been built is trained in batches, and 80 training samples are randomly trained in each batch. The weight is set as a normally distributed random number with a mean value of 0 and variance of 0.1, the bias is set as a constant of 0.1, the learning rate is set as 0.0001, and the number of training is set as 1000. Adam-Optimizer is selected to adaptively control the learning rate of each parameter in the network.

5.4. Single-Phase Ground Fault Analysis. The classification results of the voltage sag source caused by SLGF are analyzed, and only the classification results of other fault types are given. The change curve of the loss function value and the change curve of classification accuracy of the training set and test set are shown in Figure 6.

Based on the obfuscation matrix, the evaluation index of the model is calculated, each column of the obfuscation matrix represents the predicted category, and each row represents the actual category. In this paper, there are 17 types of SLGF samples, so the dimension of the confusion matrix is 17×17. To observe and draw the confusion matrix easily, as shown in Figure 7, the omitted parts are all correctly classified samples. Based on the confusion matrix, one sample of line $L2^{-}3$ ($L2^{-}3$ represents the line between node 2 and node

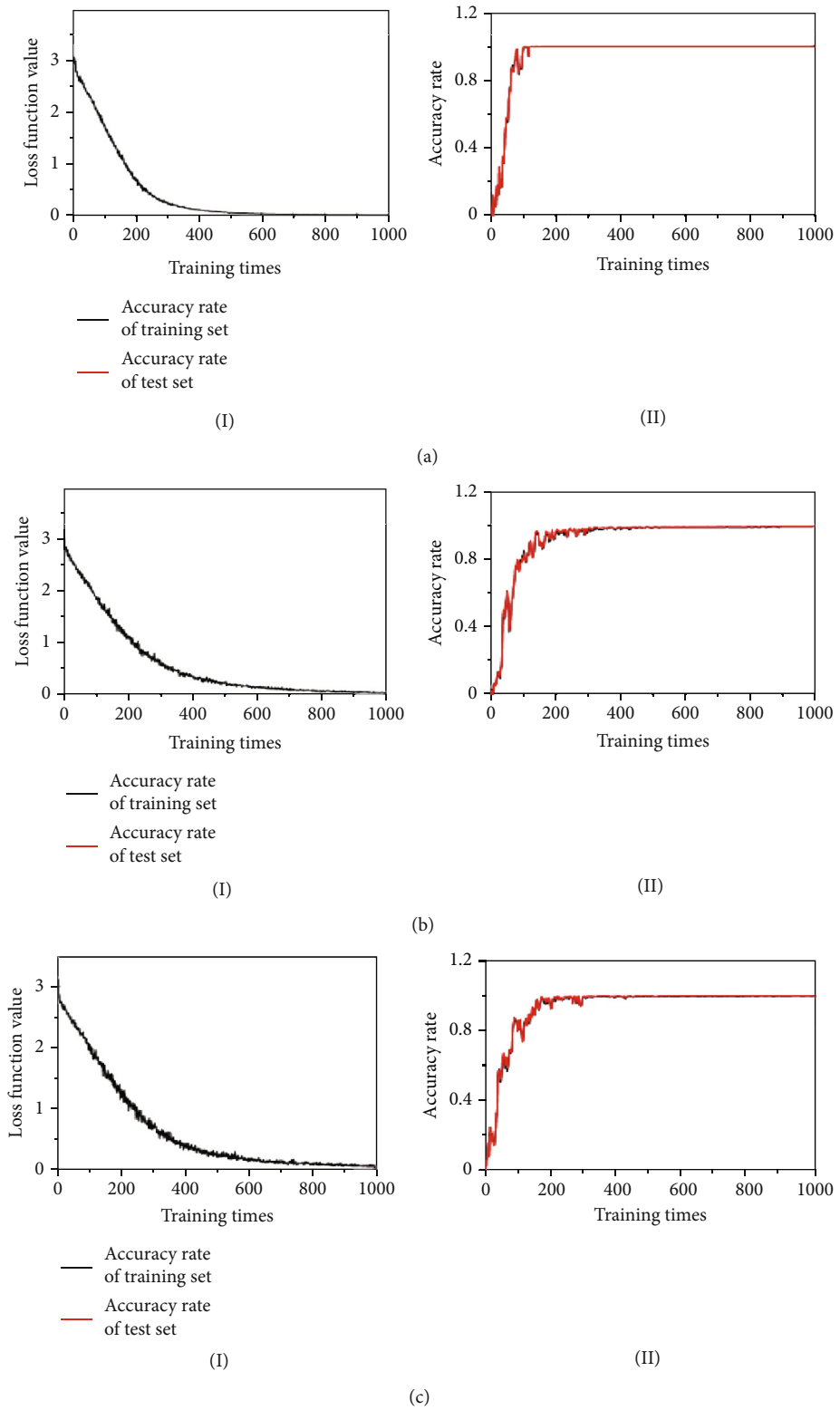


Figure 8: Training process: (a) phase-to-phase fault, (b) double line-to-ground fault, and (c) three line-to-ground fault.

3) is misclassified as line $L6^{-}13$ by the classifier and two samples of line $L^{-}4$ as line $L1^{-}2$, and the samples of other lines can be correctly classified as the real category.

The confusion matrix as shown in Figure 7 can be drawn by using the K-nearest Neighbor (KNN) and Ensemble Learning Algorithm (EL) (the confusion matrix of the classification results of the comparison classifier is omitted here due to space limitation). The respective classification indexes can be calculated according to the confounding matrices of different classifiers, as shown in Table 2.

Table 3: Classification indicator.

Fault type	Classification algorithm	Classification feature	Accuracy	Kappa
PPF	CNN	Multidimensional feature	100.00%	100.00%
	KNN	Multidimensional feature	83.24%	82.19%
		Disturbance power	88.53%	87.81%
		Equivalent impedance	76.18%	74.69%
		System slope	82.35%	81.25%
		Real part of current	82.35%	81.25%
	EL	Multidimensional feature	64.12%	61.88%
		Disturbance power	64.41%	62.19%
		Equivalent impedance	64.12%	61.88%
		System slope	64.71%	62.50%
		Real part of current	64.71%	62.50%
DLGF	CNN	Multidimensional feature	100.00%	100.00%
	KNN	Multidimensional feature	90.59%	90.00%
		Disturbance power	93.24%	92.81%
		Equivalent impedance	82.35%	81.25%
		System slope	89.41%	88.75%
		Real part of current	81.18%	80.00%
	EL	Multidimensional feature	63.82%	61.56%%
		Disturbance power	58.82%	56.25%
		Equivalent impedance	64.41%	62.19%
		System slope	64.12%	61.88%
		Real part of current	64.41%	62.19%
TLGF	CNN	Multidimensional feature	100%	100%
	KNN	Multidimensional feature	97.65%	97.50%
		Disturbance power	79.12%	77.81%
		Equivalent impedance	80.00%	78.75%
		System slope	31.18%	90.63%
		Real part of current	76.76%	75.31%
	EL	Multidimensional feature	64.12%	61.88%
		Disturbance power	35.29%	31.25%
		Equivalent impedance	35.29%	31.25%
		System slope	47.06%	43.75%
		Real part of current	64.12%	61.88%

The classification accuracy of CNN is 99.12% and the kappa coefficient is 99.06%, which are the highest among the three classifiers.

5.5. Other Short-Circuit Fault Analysis. The samples of PPF, DLGF, and TLGF are trained and classified by the same classification model. The training process of CNN is shown in Figure 8, and the classification results are shown in Table 3.

It can be seen that, when the voltage sag sources caused by different fault types are classified based on single transient characteristics, the classification accuracy of voltage sag sources caused by different fault types is seriously affected by different feature selections, different classifiers, and different fault types. The classification accuracy of KNN and EL is improved based on multidimensional features, but it cannot meet the requirements. Regardless of the voltage sag events caused by any type of fault, CNN can be used to accurately classify based on the multidimensional feature matrix, and the classification accuracy is up to 100%.

5.6. Comparison with Existing Methods. The proposed method is based on multipoint monitoring data and can be used to identify voltage sag sources in loop and radiation networks. The existing voltage sag source positioning methods based on multimonitoring points include the branch current deviation method and node voltage deviation method. Among them, the node voltage deviation method does not consider SLGF, DLGF, and TLGF faults. Therefore, the proposed method is compared with the branch current deviation method.

The branch current deviation method defines the current outflow node as the positive direction. The deviation current

Table 4: Branch current deviation table.

Line	Node	Branch current deviation	Node	Branch current deviation
$L1_2$	1	A: 26.4884 B: -0.0009 C: 0.0033	2	A: 20.5416 B: 0.0013 C: -0.0028
$L1_5$	1	A: 0.0007 B: 0.0004 C: 0.0000	5	A: 0.0007 B: 0.0004 C: 0.0000
…	…	…	…	…
$L14_{13}$	13	A: 0.0000 B: 0.0002 C: 0.0003	14	A: 0.0000 B: 0.0002 C: 0.0003

Table 5: Location result.

Fault type	SLGF	PPF	DLGF	TLGF
Accuracy	48.24%	60.59%	48.24%	48.24%

of the branch is calculated by Equation (3), and the priority is determined according to the magnitude of the deviation current, thus determining the direction of the voltage sag source. The primary SLGF on line $L1^-2$ is positioned, and the branch current deviation method is briefly described as an example.

The deviation coefficients of all branch currents are calculated, as shown in Table 4. It can be seen that, according to the priority, fault current flows through line $L1^-2$ from node 1 and through line $L1^-2$ from node 2. Therefore, it can be judged that SLGF occurs on line $L1^-2$.

Based on the branch current deviation method, the samples used for the CNN test are located for the voltage sag source, and the positioning results are shown in Table 5.

It can be seen from Table 5 that the branch current deviation method is used to locate the voltage sag caused by SLGF, DLGF, and TLGF, and the positioning accuracy is 48.24%. The accuracy of voltage sag positioning caused by PPF is 60.29%, so there is a certain gap with the method proposed in this paper. Based on the analysis of the positioning results, this method is greatly affected by the position of the power source, and the fault near the power source generates a larger fault current, so it can be positioned more accurately. Besides, the fault current in the neutral arc suppression coil system is composed of the whole system capacitive current, so this method is not applicable to the neutral arc suppression coil grounded system.

6. Conclusion

In summary, the voltage sag source location method based on the multidimensional characteristic matrix proposed in this paper can accurately locate the voltage sag caused by different fault types. Compared with the traditional voltage sag source location method based on multipoint monitoring data, it has higher accuracy and is more suitable for the voltage sag source location in the loop network.

In this paper, a locating method of the voltage sag source based on the multidimensional feature matrix was proposed; then, a multidimensional feature matrix was constructed by extracting the disturbance power and disturbance energy, the real part of equivalent impedance, the slope of system trajectory, and the real part of the current at both ends of the whole network line, and then, CNN was used to realize the voltage sag source. The IEEE14 node model was built in PSCAD\EMTDC, and a large number of waveform data obtained by simulation under different fault types was adopted to verify the proposed method in this paper, and then, it was compared with the branch current deviation method. The simulation results show that the proposed method could accurately identify the location of voltage sag sources under different fault types, and the accuracy was higher than that of traditional methods.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This paper is sponsored by State Grid Science and Technology Project (Grant No. 52153218000B).

References

[1] IEEE Std, *Recommended practice for monitoring electric power quality*, Technical report, Draft 5, 1994.

[2] Z. Yang, Z. Pingping, Y. Haojun, and L. Ganyun, "Review on cause analysis and source location for voltage sag," *Proceedings of the CSU-EPSA*, vol. 26, 2014no. 12, pp. 15–20, 2014.

[3] Z. Yan, Y. Lin, and S. Zhenguo, "Multiobjective optimal allocation of monitors for voltage sag location under observability constraint," *Transactions of China Electrotechnical Society*, vol. 34, no. 11, pp. 2375–2383, 2019.

[4] L. Xialin, L. Yajuan, and W. Zhu, "A new method to classify and identify composite voltage sag sources in distribution network," *Power System Protection &Control*, vol. 45, no. 2, pp. 131–139, 2017.

[5] L. Ganyun, J. Xiaowei, H. Sipeng et al., "Location of voltage sag source based on semi-supervised SVM," *Power System Protection and Control*, vol. 47, no. 18, pp. 76–81, 2019.

[6] A. C. Parsons, W. M. Grady, E. J. Powers, and J. C. Soward, "A direction finder for power quality disturbances based upon disturbance power and energy," *IEEE Transactions on Power Delivery*, vol. 15, no. 3, pp. 1081–1086, 2000.

[7] T. Tayjasanant, C. Li, and W. Xu, "A resistance sign-based method for voltage sag source detection," *IEEE Transactions on Power Delivery*, vol. 20, no. 4, pp. 2544–2551, 2005.

[8] C. Li, T. Tayjasanant, W. Xu, and X. Liu, "Method for voltage-sag-source detection by investigating slope of the system trajectory," *IEE Proceedings - Generation, Transmission and Distribution*, vol. 150, no. 3, pp. 367–372, 2003.

[9] J. Gao, Q. Z. Li, and J. Wang, "Method for voltage sag disturbance source location by the real current component," in *Asia-Pacific Power and Energy Engineering Conference*, IEEE Computer Society Washington, DC, USA, 2011.

[10] G. W. Chang, J.-P. Chao, H. M. Huang et al.et al., "On tracking the source location of voltage sags and utility shunt capacitor switching transients," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 2124–2131, 2008.

[11] B. Wang, W. Xu, and Z. Pan, "Voltage sag state estimation for power distribution systems," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 806–812, 2005.

[12] H. WG, D. RA, X. Y. Zhang, R. Lu, G. QQ, and X. YH, "Comparison of methods for voltage sag source detection in distribution system," *Electrical Measurement & Instrumentation*, vol. 48, no. 8, pp. 53–58, 2011.

[13] L. Haichao, O. Sen, and Z. Huaying, "A novel location method of power grid voltage sag source with BP neural network," *Journal of Electric Power Science and Technology*, vol. 2, 2017.

[14] L. Ganyun and Y. Wu, "Optimization comprehensive criterion for voltage sag source location," *Power System Protection and Control*, vol. 5, pp. 66–71, 2013.

[15] S. Weimeng, *Influence Assessment and Location Research of Voltage Sags Based Intelligent Algorithm*, Zhejiang Normal University, 2011.

[16] K. Wang, *Research on Voltage Sag Source Location Method Based on Comprehensive Criterion*, China University of Mining and Technology, 2018.

[17] L. Yingying, T. Wang, and F. Dandan, "Fault source localization method based on multiple criteria for voltage dips," *Proceedings of the CSEE*, vol. 35, no. 1, pp. 103–111, 2015.

[18] Z. Chenyu, S. Mingming, F. Zhong, J. Zheng, and Y. Xiaodong, "Research on voltage sag event classification and short circuit type identification," *Electric Power Engineering Technology*, vol. 37, no. 2, pp. 102–107, 2018.

[19] Y. Baocai, W. Wang, and L. Wang, "Review of deep learning," *Journal of Beijing University of Technology*, vol. 1, pp. 48–59, 2015.

[20] C. Zheng, X. Kong, G. Jiahua et al., "Traveling wave head detection algorithm based on wavelet multiscale information fusion," *Smart Power*, vol. 47, no. 5, pp. 97–102, 2019.

[21] C. Wei, J. He, and P. Xiping, "Classification for power quality disturbance based on phase-space reconstruction and convolution neural network," *Power System Protection and Control*, vol. 46, no. 14, pp. 87–93, 2018.

[22] Y. Zhao, H. Li, S. Wan et al., "Knowledge-aided convolutional neural network for small organ segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1363–1373, 2019.

[23] 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*, vol. 32, no. 9, pp. 4387–4415, 2020.

[24] Y. Zhang, J. Pan, L. Qi, and Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *Future Generation Computer Systems*, vol. 114, pp. 336–348, 2021.

[25] T.-T. Goh, Z. Xin, and D. Jin, "Habit formation in social media consumption: a case of political engagement," *Behaviour & Information Technology*, vol. 38, no. 3, pp. 273–288, 2019.

[26] D. Jin, S. Shi, Y. Zhang, H. Abbas, and T.-T. Goh, "A complex event processing framework for an adaptive language learning system," *Future Generation Computer Systems*, vol. 92, pp. 857–867, 2019.

[27] G. Jike, C. Dong, W. Wang, C. Zuqin, C. Guorong, and L. Can, "On the fire-disaster classification based on the improved naive Bayesian algorithm," *Journal of Safety and Environment*, vol. 19, no. 4, pp. 1122–1127, 2019.

[28] S. K. Goudos, P. D. Diamantoulakis, and G. K. Karagiannidis, "Multi-objective optimization in 5G wireless networks with massive MIMO," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2346–2349, 2018.

[29] S. K. Goudos, Z. D. Zaharis, and K. B. Baltzis, "Particle swarm optimization as applied to electromagnetic design problems," *International Journal of Swarm Intelligence Research*, vol. 9, no. 2, pp. 47–82, 2018.

[30] Z. Gao, H. Xue, and S. Wan, "Multiple discrimination and pairwise CNN for view-based 3D object retrieval," *Neural Networks*, vol. 125, pp. 290–302, 2020.

[31] Y. Zhang, Y. Jin, J. Chen, S. Kan, Y. Cen, and Q. Cao, "PGAN: part-based nondirect coupling embedded GAN for person reidentification," *IEEE Multimedia*, vol. 27, no. 3, pp. 23–33, 2020.

[32] S. Wan, X. Xu, T. Wang, and Z. Gu, "An intelligent video analysis method for abnormal event detection in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2020.

[33] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, vol. 398, pp. 520–530, 2020.

[34] J. Jia, Q. Ruan, Y. Jin, G. An, and S. Ge, "View-specific subspace learning and re-ranking for semi-supervised person reidentification," *Pattern Recognition*, vol. 108, article 107568, 2020.

[35] Z. Gao, Y. Li, and S. Wan, "Exploring deep learning for view-based 3D model retrieval," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1, pp. 1–21, 2020.

[36] Y. Xi, Y. Zhang, S. Ding, and S. Wan, "Visual question answering model based on visual relationship detection," *Signal Processing: Image Communication*, vol. 80, p. 115648, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6664324, 14 pages
https://doi.org/10.1155/2021/6664324

Research Article

Edge Sensing-Enabled Multistage Hierarchical Clustering Deredundancy Algorithm in WSNs

Rongbo Zhu,[1] **Mai Yu**,[1] **Yuanli Li**,[1] **Jun Wang**,[1] **and Lu Liu**[2]

[1]*College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China*
[2]*School of Informatics, University of Leicester, Leicester, UK*

Correspondence should be addressed to Rongbo Zhu; rbzhu@mail.scuec.edu.cn

Received 25 November 2020; Revised 31 January 2021; Accepted 22 February 2021; Published 12 March 2021

Academic Editor: Shaohua Wan

Due to the defects caused by limited energy, storage capacity, and computing ability, the increasing amount of sensing data has become a challenge in wireless sensor networks (WSNs). To decrease the additional power consumption and extend the lifetime of a WSN, a multistage hierarchical clustering deredundancy algorithm is proposed. In the first stage, a dual-metric distance is employed, and redundant nodes are preliminarily identified by the improved *k*-means algorithm to obtain clusters of similar nodes. Then, a Gaussian hybrid clustering classification algorithm is presented to implement data similarity clustering for edge sensing data in the second stage. In the third stage, the clustered sensing data is randomly weighted to deduplicate the spatial correlation data. Detailed experimental results show that, compared with the existing schemes, the proposed deredundancy algorithm can achieve better performance in terms of redundant data ratio, energy consumption, and network lifetime.

1. Introduction

Wireless sensor networks (WSNs) are common in people's lives and are widely used in various fields [1, 2]. WSNs are deployed in different areas to monitor environments and objects, such as temperature, humidity, and seismic events [3, 4]. To obtain accurate sensing data for events, a large number of sensors are utilized to collect the edge sensing data and transmit the data to an aggregation/sink node in a high-frequency manner. In general, edge sensing data have a high spatial-temporal correlation and contain considerable redundant information [5, 6]. Additionally, the transmission of redundant data leads to unnecessary energy consumption and bandwidth costs, which increase the overhead and decrease WSN lifetimes. Therefore, reducing redundant data and the transmission energy consumption to extend network lifetimes becomes a key issue in WSNs.

To reduce redundant data effectively, the existing work concentrates on two aspects: optimizing sensing data and predicting sensing data. On the one hand, the former is aimed at reducing redundant sensing data with some optimized schemes. Considering the constrained resources in WSNs, a spatial-temporal correlation data reduction scheme was proposed to determine the optimal sampling strategy for the deployed sensor nodes (SNs) [7]; the strategy reduces the overall sampling/transmission rates while preserving the quality of the data. Considering that the data volume increases with unexpected ratios in WSNs, an integrated divide and conquer method with an enhanced *k*-means scheme was proposed [8], which removes redundant data from the collected measures. To save the limited energy of WSNs, a data transmission (Dat) protocol, which can reduce the data transmission cost inside each sensor node by removing redundant data to save energy while maintaining a suitable level of accuracy in the received readings at the sink, was presented [9]. To conserve energy and enhance the lifetime of a WSN, reducing the amount of data communicated by exploiting the temporal and spatial correlations of the sensed data is a suitable approach. An energy-efficient semantic clustering model was proposed to mitigate the high-energy consumption problem in a clustered WSN [10]. To reduce the energy consumed during data transmission, an adaptive data reduction method, which is based on a convex combination of two decoupled least-mean-square windowed filters, was proposed [11].

On the other hand, the prediction-based scheme tries to reduce sensing data with forecasting schemes. To improve data processing efficiency, a distributed data prediction model based on least squares, which tries to use a data prediction-based filtering scheme, was proposed to decrease transmission data [12]. Alduais et al. [13] presented an updating frequency metric, which is defined as the frequency of updating the model reference parameters during data collection, to evaluate the performance of different multivariate data reduction models for WSNs. To schedule data communications between SNs and a sink to reduce power usage with the aim of maximizing the network lifetime, a prediction-based data communication scheme, which utilizes the hierarchical least-mean-square adaptive filter to predict the measured values both at the source and at the sink, was presented [14].

Although the schemes mentioned above provide efficient solutions to reduce redundant sensing data in WSNs, the following defects still need to be addressed comprehensively. Firstly, a large range of edge sensing data and errors in local best values can lead to local characteristics being lost. Then, the errors of sensing data will result in a similarity threshold failure problem. And the existing distance-based correlation reducing redundant sensing data schemes, which only consider the spatial corrections of sensing event and omit the temporal correlations of sensing data, tend to degrade the accuracy of sensing data. Furthermore, prediction-based schemes require relatively long-term data sensing and processing abilities, which increase the burden of resource-limited sensors and decrease the lifetime of WSNs. Hence, it is necessary to consider both spatial and temporal correlations and location and data similarity clustering to decrease the sensing data transmission and processing. Focusing on the issue mentioned above, this paper explores the sensing data deredundancy problem to decrease energy consumption and extend the lifetime of a WSN.

The main contributions of this paper are summarized as follows:

(1) A multistage hierarchical clustering similarity deredundancy (MHCSD) algorithm is proposed to reduce the power consumption and extend the lifetime of a WSN. MHCSD considers both spatial and temporal correlations and location and data similarity clustering to overcome the accuracy degradation of sensing data

(2) A dual-metric distance is employed in the first stage, and an improved k-means algorithm is proposed to judge the similarity of nodes based on the dual-metric distance in sinks. A Gaussian hybrid clustering algorithm is presented to judge the similarity of edge sensing data within the same cluster and can improve the similarity accuracy and the deredundancy ratio. The clustered sensing data are randomly weighted to further deduplicate the spatial correlation data in the third stage

The remainder of this paper is organized as follows. Related work is explored in Section 2. Section 3 presents the proposed multistage hierarchical clustering deredundancy algorithm in WSNs. Section 4 shows the experimental results, which verify the proposed scheme. The paper is concluded in Section 5 finally.

2. Related Work

Although a large amount of application-specific data are generated in WSNs, most of the sensing data detected by sensors are redundant. Processing and transmitting massive superfluous data can lead to additional power consumption and greatly decrease network lifetime [15, 16]. To improve data processing performance, a path merging protocol, which supports partial discrete wavelet transform-based compression schemes to reduce redundant data transmission in a significant manner through the appropriate aggregation of data packets from merging paths, was proposed in [17]. To manage energy-efficient data collections in WSNs, a data-aware energy conservation scheme and prediction-based data collection framework were proposed to reduce data transmission [18], where the inherent correlation between the consecutive observations of SNs and the data similarity measures between the neighboring SNs are utilized.

Considering that the data volume in WSNs is quickly increasing, a hybrid-stream big data analytics model, which utilizes a multidimensional convolutional neural network (CNN), minimal correlation model, and minimal redundancy model to optimize data processing, is proposed to perform big data analysis [19]. To provide a complete description of an environment and make a robust decision, a redundancy removal strategy, which mines the spatial and temporal data from collected data to select the appropriate information before forwarding to a base station or a cluster head (CH) in a WSN, is proposed [20]. To avoid generating, transmitting, and storing unwanted data from redundant messages, an immunization-based redundancy elimination scheme, which independently selects the correct number of acknowledgment frames distributed to respond to variations in the amount of redundant data in a dynamic fashion, was proposed [21]. An image fusion method was proposed based on histogram similarity and multiview weighted sparse representations [22].

By introducing histogram similarity, different weights are given to low-resolution high-frequency components and source image high-frequency components, and complementary information is effectively used. Diwakaran et al. [18] used the inherent correlations between the continuous observations of SNs and the data similarity measures of adjacent SNs to reduce data transmission. A new model based on monkey tree search behavior inspired by fauna was explored in [23], and the fuzzy reasoning mechanism was used to complete data collection and dissemination. Rida et al. [24] utilized data aggregation techniques based on the Euclidean distance to reduce similar data. Lin et al. [25] proposed a semantic data annotation method based on semantics. A data clustering method, which groups homogeneous data into clusters and then performs data reduction by selecting the average value of each cluster, was proposed based on histograms for data reduction [26].

Additionally, to address the problem of redundant data collected by sensors, data aggregation and semaphore processing based on similar functions are applied in WSNs, and SNs are aggregated with a palm tree method [27]. Wan et al. [28] proposed a similar sensory data aggregation scheme based on fuzzy c-means. A spatial-temporal correlation search mechanism between SNs based on the Euclidean distance is proposed [29]. An energy-saving redundant traffic handling scheme, which utilizes short beacon information to process redundant packets generated in area-based routing, was presented in [30]. The parameter estimation problem was considered in [31], and two censoring algorithms were proposed to enable SNs to transmit sampled data based on local decision-making. The dual prediction scheme is used to reduce the transmission between cluster nodes and CHs, while the data compression scheme is used to reduce the traffic between CHs and sink nodes [32]. A low redundancy data acquisition scheme, which selects some nodes for data detection and transmits less data to CHs, was proposed based on matrix completion [33]. To reduce transmitted data, a differential data processing (DDP) method was proposed in [34].

Although there are many effective deredundancy processing schemes in WSNs, the following limitations still need to be addressed. Data correlation analysis does not consider homologous data, which can result in a lower deredundancy ratio and loss of local characteristics. Furthermore, unconscionable deredundancy can degrade the accuracy of sensing data. Focusing on filling this gap, this paper proposes a multiphase hierarchical clustering similarity deredundancy algorithm to overcome the limitations mentioned above.

3. Proposed Scheme

3.1. System Model. There is a set $S = \{s_1, s_2, \cdots, s_n\}$ composed of n SNs. And the edge sensor nodes will collect data. The system model is shown in Figure 1, and Table 1 lists the notation that we use in the paper.

The sink calculates the similar distances between nodes according to the coordinates of the nodes and divides the nodes into K clusters according to the similar distances $S = \{C_1, C_2, C_3, \cdots, C_K\}$, where $C_i \cap C_j = \varnothing (i \neq j)$. CH_i of each cluster collects the sensing data generated by the nodes in the cluster at time t_j as set $D_{\mathrm{CH}_h}(t_j) = \{x_1(t_j), x_2(t_j), \cdots, x_n(t_j)\}$. Gaussian mixed clustering is adopted to classify the collected data into similar clusters and then classifies the nodes in the cluster as $C_i = \{C_{i,1}, C_{i,2}, C_{i,3}, \cdots, C_{i,m}\}$, $C_{i,j} = \{s_{i,j,1}, s_{i,j,2}, \cdots, s_{i,j,k}\}$, where $1 \leq i \leq K, 1 \leq j \leq m$.

3.2. Deredundancy Algorithm. The proposed MHCSD algorithm includes three stages: the local clustering stage, the similar data clustering stage, and the data deredundancy processing stage. In addition, the framework of MHCSD is shown in Figure 2. In the first stage, the sink will perform the improved k-means clustering algorithm, which clusters similar nodes according to the spatial position coordinates of the nodes. Then, in the second stage, CHs adopt the Gaussian hybrid clustering algorithm to further seek similar clusters. In the third stage, based on the maximum time threshold, the SNs utilize an adaptive step length in the data deredundancy scheme (TCDA) to eliminate duplicate sensing data with spatial-temporal correlations.

3.2.1. Similarity of SNs. In the first stage, the node similarity analysis is performed according to the node position coordinates in the sink. To delete duplicate edge sensing data effectively, local clustering needs a precise similarity measure among nodes and sensing data. Among various distance metrics, the Euclidean distance may be the most commonly used in data processing. However, the Euclidean distance only describes the amplitude difference between two eigenvectors, and the Euclidean distance of two feature vectors with different shapes may be smaller than that of feature vectors with similar shapes. To overcome the defect in the Euclidean distance, a dual-metric similarity distance $D(i, j)$ is employed:

$$D(i, j) = D_E(i, j) + \beta D_P(i, j), \tag{1}$$

where $D_E(i, j)$ is the Euclidean distance, $D_P(i, j)$ is the Pearson correlation distance, and β is a scale factor that indicates the influence of $D_P(i, j)$ on the weight of $D(i, j)$. In addition, we have

$$D_E(i, j) = \sqrt{\sum_{h=1}^{h} \left(l_{i,h} - l_{j,h}\right)^2}, \tag{2}$$

$$D_P(i, j) = \frac{1}{2}\left(1 - \frac{\sum_{h=1}^{h}\left(l_{i,h} - \bar{l}_{i,h}\right)\left(l_{j,h} - \bar{l}_{j,h}\right)}{\sqrt{\sum_{h=1}^{h}\left(l_{i,h} - \bar{l}_{i,h}\right)^2 \sum_{h=1}^{h}\left(l_{j,h} - \bar{l}_{j,h}\right)^2}}\right). \tag{3}$$

The dual-metric similarity distance $D(i, j)$ meets three distance characteristics: positivity, symmetry, and reflexivity. In terms of $D(i, j)$, any active feature vector pair can be compared from both the amplitude of the Euclidean distance and the change in the shape of the related distance.

For $S = \{s_1, s_2, \cdots, s_n\}$, the spatial position coordinate l_i of node s_i is (x_i, y_i). The sink performs the improved k-means algorithm as shown in Algorithm 1.

According to coordinate position set $L = \{l_1, l_2, \cdots, l_n\}$, n nodes are classified in K disjoint subsets C_i. $C = \{C_1, C_2, \cdots, C_K\}$ and $C_1 \cup C_2 \cup \cdots \cup C_K = S$, where $C_i \neq \varnothing$ and $C_i \cap C_j = \varnothing$, $i \neq j$. In addition, the minimum squared error e is defined as

$$e = \sum_{i=1}^{K} \sum_{l \in C_i} \|l - \bar{\mu}_i\|_2^2, \tag{4}$$

where $\bar{\mu}_i = (1/|C_i|)\sum_{l \in C_i} l$ is the mean vector of cluster C_i.

3.2.2. Similarity of Sensing Data. After clustering the similar nodes by the spatial positions in the first stage, to refine redundant judgments of nodes, the Gaussian hybrid

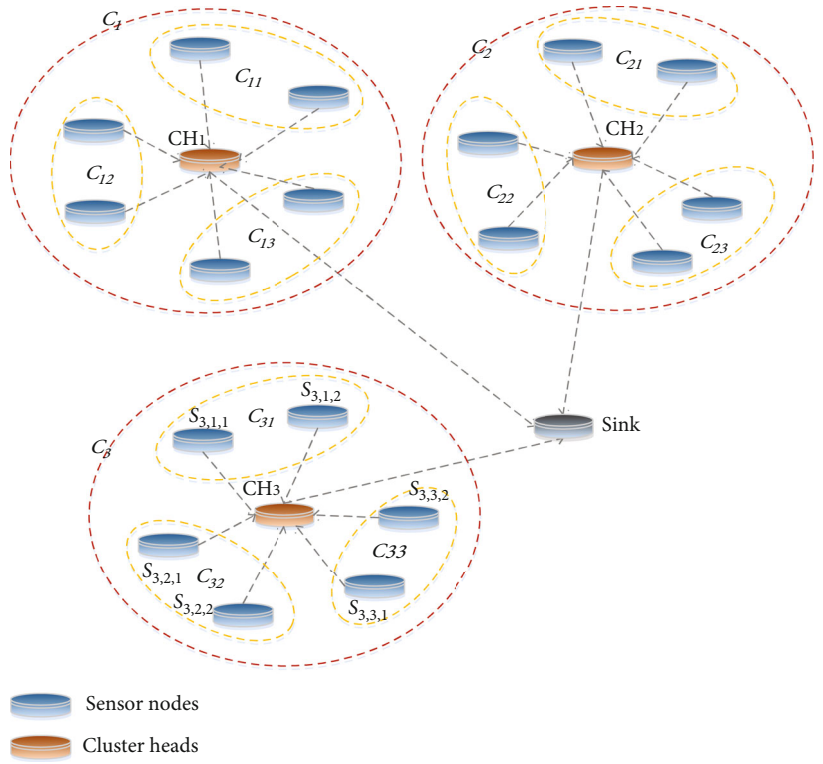


Figure 1: System model.

clustering algorithm is adopted in the second stage and is shown in Algorithm 2.

To further analyze the similarity of the data collected simultaneously within cluster C_i, the probabilistic model is used to analyze and describe the prototype data in Gaussian hybrid clustering. The cluster division is mainly determined by the posterior probability corresponding to the prototype. The Gaussian distribution is defined as the random variable x in the n-dimensional sample space X, and its probability density function $p(x)$ is defined as

$$p(x) = \frac{1}{(2\pi)^{n/2}|\aleph|^{1/2}} e^{-1/2(x-\mu)^T \aleph^{-1}(x-\mu)}, \tag{5}$$

where μ represents the n-dimensional mean vector and $\aleph$ denotes the $n \times n$ covariance matrix. Since the Gaussian distribution is determined by the mean vector μ and the covariance matrix $\aleph$, for the convenience of description, the probability density function for the dependence of the Gaussian distribution on the corresponding parameters is expressed as $p(x \mid \mu, \aleph)$. The Gaussian mixture distribution p_M is

$$p_M = \sum_{i=1}^{K_1} \alpha_i \bullet p(x \mid \mu_i, \aleph_i), \tag{6}$$

where μ_i and $\aleph_i$ are the parameters of the ith Gaussian mixed component, $\alpha_i > 0$ is the corresponding mixing coefficient, and $\sum_{i=1}^{K_1} \alpha_i = 1$. p_M consists of K_1 mixed components, and each mixed component corresponds to a Gaussian distribution.

For S composed of K clusters, data generated by a cluster can be expressed as a set $X = \{X_1, X_2, \cdots, X_n\}$, and $X_i = \{x_i(t_1), x_i(t_2), \cdots, x_i(t_2)\}$, where $1 \le i \le n$ is the set of time series generated by the sensor node s_i every T seconds. In the WSN, each CH continues to classify the correlated data of the nodes in the cluster, and through the Gaussian hybrid clustering algorithm, the data collection $D_{\mathrm{CH}_h}(t_j) = \{x_1(t_j), x_2(t_j), \cdots, x_z(t_j)\}$ in similar clusters of the same spatial nodes is simultaneously divided into K_1 clusters, where $1 \le j \&\& 1 \le h \le K_1$.

It is assumed that the random variable $z_{j_1} \in \{1, 2, \cdots, K_1\}$ represents the Gaussian mixture component of the sensing data $x_{j_1}(t_j)$ of node j_1. The prior probability $P(z_{j_1} = i)$ of z_{j_1} corresponds to $\alpha_i (i = 1, 2, \cdots, K_1)$. According to Bayes' theorem, the posterior distribution of z_{j_1} corresponds to

$$\begin{aligned} p_M\left(z_{j_1} = i \mid x_{j_1}(t_j)\right) &= \frac{P\left(z_{j_1} = i\right) \bullet p_M\left(x_{j_1}(t_j) \mid z_{j_1} = i\right)}{p_M\left(x_{j_1}(t_j)\right)} \\ &= \frac{\alpha_i \bullet p\left(x_{j_1}(t_j) \mid \mu_i, \aleph_i\right)}{\sum_{l=1}^{k} \alpha_l \bullet p\left(x_{j_1}(t_j) \mid \mu_l, \aleph_l\right)}. \end{aligned} \tag{7}$$

$p_M(z_{j_1} = i \mid x_{j_1}(t_j))$ is expressed as sample $x_{j_1}(t_j)$ generated by the ith Gaussian mixture composition of the a posteriori probability, expressed as $\gamma_{j_1 i} (i = 1, 2, \cdots, K_1)$.

Table 1: Notation.

Symbol	Description
K	Number of similar clusters
β	The effect of $D_P(i,j)$ on the weight of $D(i,j)$, weight scaling factor
n	Node size
m	Number of similar clusters
K_1	Number of data similar clusters
α_i	Gaussian mixture coefficient
z_{j_1}	Gaussian mixture components
$\gamma_{j_1,i}$	The posterior probability of the Gaussian mixture
ρ_{j_1}	Cluster markers of sample $x_{j_1}(t_j)$
μ_i	The average of the components
ρ	Lagrange multiplier
β_i	Random weighting factor
e	Minimizing the squared error in clustering of k-means
E_{elec}	Power consumption of a circuit for sending or receiving data
$\varepsilon_{mp}, \varepsilon_{fs}$	Energy consumption of signal amplifiers
E_P	Energy consumption per unit of data
$D_E(i,j)$	Euclidean distance
$D_P(i,j)$	Pearson correlation distance
$D(i,j)$	Spatial similarity distance
$\bar{\mu}_i$	The mean of cluster C_i
$\aleph$	$n \times n$ covariance matrix
ρ_{j_1}	Cluster markers
$D_{C_{i,j_1}}(t_j)$	The random weighted deredundancy of sensing data in C_{i,j_1}

After the Gaussian mixture distribution, the cluster becomes the sample set $D_{\text{CH}_h}(t_j)$ divided into K_1 subclusters and expressed as set $C = \{C_{i,1}, C_{i,2}, C_{i,3}, \cdots, C_{i,K_1}\}$ $(0 < i \le K_1)$; the cluster markers ρ_{j_1} of each sample $x_{j_1}(t_j)$ are defined as follows:

$$\rho_{j_1} = \underset{i\in(1,2,\cdots,3)}{argmax} \gamma_{j_1,i}. \tag{8}$$

We can solve the parameters $\{(\alpha_i, \mu_i, \aleph_i) \mid 1 \le i \le K_1\}$ as

$$LL(D) = \ln\left(\prod_{j_1=1}^{m} p_M\left(x_{j_1}(t_j)\right)\right) = \sum_{j_1=1}^{m} \ln\left(\sum_{i=1}^{K_1} \alpha_i \bullet p\left(x_{j_1}(t_j) \mid \mu_i, \aleph_i\right)\right). \tag{9}$$

The expectation maximization algorithm is used for the iterative optimization solution. To maximize equation (8) by $\partial LL(D)/\partial \mu_i = 0$, we use

$$\sum_{j_1=1}^{m} \frac{\alpha_i \bullet p\left(x_{j_1}(t_j) \mid \mu_i, \aleph_i\right)}{\sum_{l=1}^{K1} \alpha_l \bullet p\left(x_{j_1}(t_j) \mid \mu_i, \aleph_i\right)} \left(x_{j_1}(t_j) - \mu_i\right) = 0, \tag{10}$$

and with $\gamma_{j_1,i} = p_M(z_{j_1} = i \mid x_{j_1}(t_j))$, we can obtain

$$\mu_i = \frac{\sum_{j_1=1}^{m} \gamma_{j_1,i} x_{j_1}(t_j)}{\sum_{j_1=1}^{m} \gamma_{j_1,i}}, \tag{11}$$

where μ_i is the mean of each mixed component and can be estimated by the weighted average of samples. The sample weight is the posterior probability $\gamma_{j_1 i}$ of each sample belonging to the component. Similarly, from $\partial LL(D)/\partial \aleph_i = 0$, we can obtain

$$\aleph_i = \frac{\sum_{j_1=1}^{m} \gamma_{j_1,i} \left(x_{j_1}(t_j) - \mu_i\right)\left(x_{j_1}(t_j) - \mu_i\right)^T}{\sum_{j_1=1}^{m} \gamma_{j_1,i}}. \tag{12}$$

For mixed coefficient α_i, in addition to maximizing $LL(D)$, it needs to satisfy $\alpha_i \ge 0$ and $\sum_{i=1}^{K_1} \alpha_i = 1$.

The Lagrange form of $LL(D)$ is

$$LL(D) + \rho\left(\sum_{i=1}^{K_1} \alpha_i - 1\right), \tag{13}$$

where ρ is the Lagrange multiplier. The derivative of equation (12) with respect to α_i is 0, and

$$\sum_{j_1=1}^{m} \frac{p\left(x_{j_1}(t_j) \mid \mu_i, \aleph_i\right)}{\sum_{l=1}^{K1} \alpha_l \bullet p\left(x_{j_1}(t_j) \mid \mu_l, \aleph_l\right)} + \rho = 0. \tag{14}$$

Both sides are multiplied by α_i, all of the components of the mixture are summed, $\rho = -m$, and

$$\alpha_i = \frac{1}{m} \sum_{j_1=1}^{m} \gamma_{j_1,i}; \tag{15}$$

namely, the mixing coefficient of each Gaussian component is determined by the average posterior probability of the sample.

(1) Elimination of Similar Data. According to the result of cluster set $C_1 = \{C_{1,1}, C_{1,2}, C_{1,3}, \cdots, C_{1,K_1}\}$ in the second stage, the CH randomly weights the data generated by the nodes in the cluster with similar data simultaneously, and the TCDA algorithm is proposed to perform time-dependent deredundancy. CHs finally transmit the deredundant data $D_{C_{i,j_1}}(t_j)$ to the sink, and we have

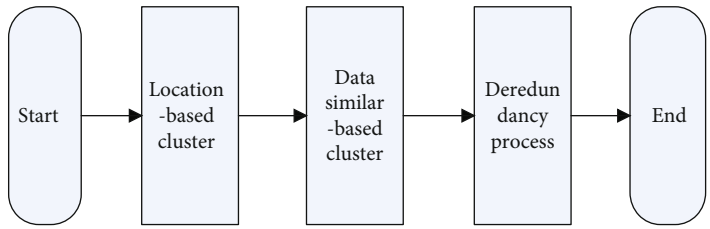


Figure 2: The framework of the proposed MHCSD.

Input: $L = \{l_1, l_2, \cdots, l_n\}$; K
Output: Result of cluster division $C = \{C_1, C_2, \cdots, C_K\}$
Randomly choses K sample from L as initial mean vector $\{\mu_1, \mu_2, \cdots, \mu_K\}$
Repeat
  $C_i \longleftarrow \varnothing (1 \le i \le K)$
  For $j \longleftarrow 1, 2, \cdots, n$ **do**
    Solving the distance l_j and all kinds of $\mu_i (1 \le i \le K)$: $D(i, j) \longleftarrow D_E(i, j) + \beta D_P(i, j)$
    The cluster marker of l_j is determined according to the nearest mean vector: $\tau_j \longleftarrow \arg \min_{i\epsilon\{1,2,\cdots,K\}} D(i, j)$
    $C_{\tau_j} \longleftarrow C_{\tau_j} \cup \{l_j\}$
  End For
  For $i \longleftarrow 1, 2, \cdots, K$ **do**
    $\mu_i{}' \longleftarrow (1/|C_i|)\sum_{l\in C_i} l$
    If $\mu_i{}' \ne \mu_i$ then
      $\mu_i{}' \longleftarrow \mu_i$
    Else
      Keeping the current mean vector not to change
    End If
  End For
Until The current mean vectors are not updated

Algorithm 1: ImpkMeans(L, K).

Input: $D_{CH_h}(t_j) = \{x_1(t_j), x_2(t_j), \cdots, x_z(t_j)\}$; K_1
Output: Result $C_1 = \{C_{1,1}, C_{1,2}, C_{1,3}, \cdots, C_{1,K_1}\}$
Repeat
  For $j_1 \longleftarrow 1, 2, 3, \cdots, z$ **do**
    According to (4.7) calculating the posterior probability generated by the mixed components $x_{j_1}(t_j)$,
    $\gamma_{j_1,i} \longleftarrow p_M(z_{j_1} = i \mid x_{j_1}(t_j))(1 \le i \le K_1)$
  End For
  For $i \longleftarrow 1, 2, 3, \cdots, K_1$ do
    $\mu_i{}' \longleftarrow (\sum_{j_1=1}^{m} \gamma_{j_1,i} x_{j_1}(t_j) / \sum_{j_1=1}^{m} \gamma_{j_1,i})$
    $\aleph_i{}' \longleftarrow (\sum_{j_1=1}^{m} \gamma_{j_1,i} (x_{j_1}(t_j) - \mu_i{}')(x_{j_1}(t_j) - \mu_i{}')^T / \sum_{j_1=1}^{m} \gamma_{j_1,i})$
    $\alpha_i{}' \longleftarrow (1/m)\sum_{j_1=1}^{m} \gamma_{j_1,i}$
  End For
  Update model parameters:$\{(\alpha_i, \mu_i, \aleph_i) \mid 1 \le i \le K_1\} \longleftarrow \{(\alpha_i{}', \mu_i{}', \aleph_i{}') \mid 1 \le i \le K_1\}$
Until Satisfy the stop condition
  $C_i \ne \varnothing (1 \le i \le K_1)$
For $j_1 \longleftarrow 1, 2, 3, \cdots, z$ do
  According to (4.7) determining cluster markets ρ_{j_1} of $x_{j_1}(t_j)$
  $C_{1,\rho_{j_1}} \longleftarrow C_{1,\rho_{j_1}} \cup \{x_{j_1}(t_j)\}$
End For

Algorithm 2: GMM$(D_{CH_h}(t_j), K_1)$.

Input: L ; K; $D_{CH_h}(t_j)$; K_1
Output: Deredundancy data
Resut1=ImpkMeans(L, K)
Rerult2=GMM($D_{CH_h}(t_j), K_1$)
Result3 is that CH randomly weights the data generated by the nodes in the cluster with similar data.
TCDA algorithm will process Result3 data.

Algorithm 3: HMDA($L, K, D_{CH_h}(t_j), K_1$).

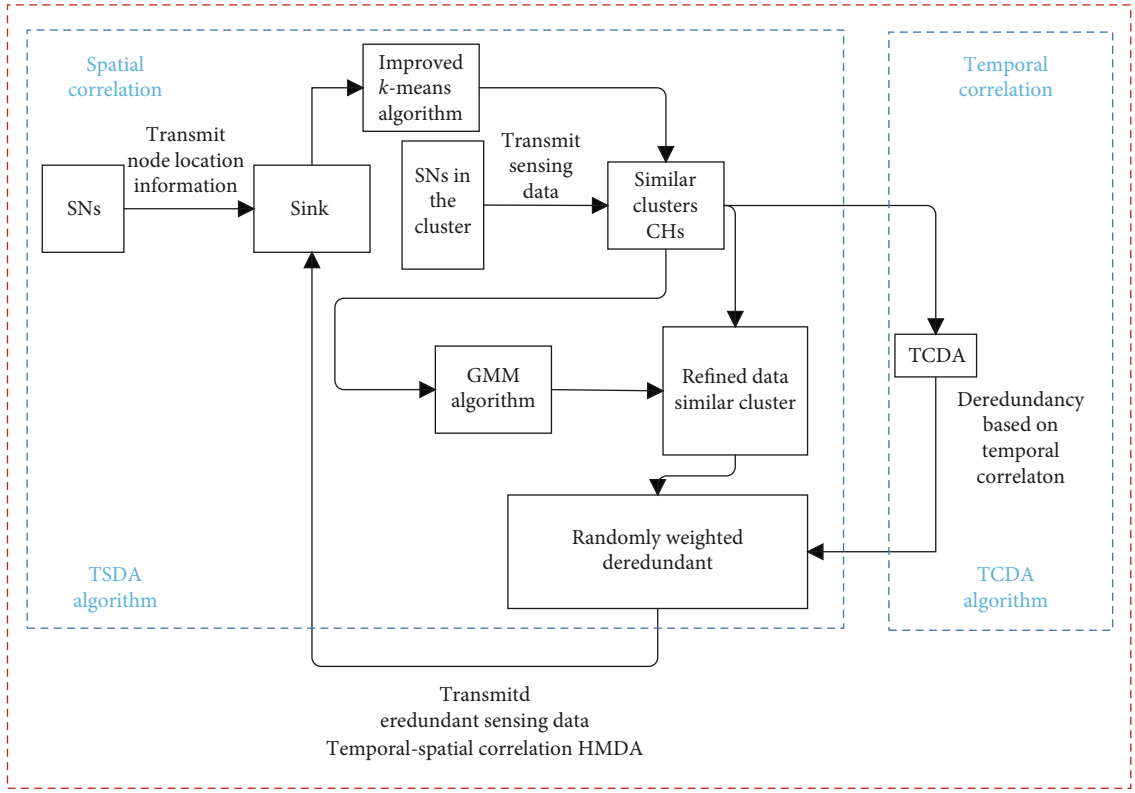


Figure 3: The flow chart of HMDA.

$$D_{C_{i,j_1}}(t_j) = \beta_1 x_w(t_j) + \beta_2 x_a(t_j) + \cdots + \beta_v x_b(t_j), \quad (16)$$

where $\beta_1, \beta_2, \cdots, \beta_v$ are weighting factors; $\sum_{j=1}^{v} \beta_j = 1$; $x_w(t_j)$, $x_a(t_j), \cdots, x_b(t_j)$ are the sensing data generated from nodes $s_w, s_a, \cdots, s_b$ at t_j; and $s_w, s_a, \cdots, s_b \in C_{ij_1}$ &&0 < $w, a, \cdots, b \le n$ &&0 < $i \le K$&&0 < $j_1 \le K_1$.

3.2.3. HMDA Algorithm. To reduce redundant data in WSNs, a hybrid multistage deredundancy algorithm (HMDA), which combines MHCSD and TCDA to reduce redundant data comprehensively, is proposed based on spatial-temporal correlations, as shown in Algorithm 3.

The MHCSD algorithm reduces redundancy in terms of spatial correlations, and the TCDA algorithm further reduces redundant data in terms of temporal correlations. TCDA fully considers the following factors in the process of deduplication: when the range of data variation is large, there is a large error in the local maximum or minimum value and a missing local eigenvalue, and when the data fluctuation is stable, the data similarity threshold cannot work effectively. Considering the ratio of deduplication, TCDA guarantees the timeliness of the sensing data with a maximum time threshold to prevent a failure in the data similarity threshold. Furthermore, an adaptive step size mechanism is proposed to reduce the complexity of calculation and energy consumption. Hence, HMDA reduces network energy consumption and extends the lifetime of a WSN simultaneously. In addition, the flow chart of HMDA is shown in Figure 3.

3.3. Performance Analysis

3.3.1. Algorithm Complexity. In the first stage, the sink aggregates through all node positions, classifies all nodes, and assumes that model training requires f_1 cycles. In the first step, the position set and classification number K of n nodes are input, and the time complexity is $O(n + 1)$; in the second step, K samples are randomly selected as the initial mean

vector, and the time complexity is $O(K)$; in the third step, the distance between each sample and K means is calculated, and the time complexity is $O(f_1 \times n \times K)$; in the fourth step, the mean vector is updated, and the time complexity is $O(f_1 \times K \times q)\&\&(1 < q < n)$; in the fifth step, the cluster division results are output, and the time complexity is $O(K \times q)\&\&(1 < q < n)$. In the second stage, the CH performs a data similarity analysis of the cluster data generated at each moment, assuming that model training requires f_2 cycles. The first step is to input the sensing data of z nodes and similarity number K_1, and the time complexity is $O(z+1)$; the second step is to calculate the posterior probability generated by each mixed component, and the time complexity is $O(f_2 \times z)$; the third step is to calculate each model parameter, and the time complexity is $O(f_2 \times K_1)$; the fourth step is to calculate the cluster tag's classification, and the calculation complexity is $O(z \times K_1)$; the fifth step is to output K_1 classification clusters, and the time complexity is $O(z \times K_1)$. In the third stage, the CHs perform random weighted transmission to reduce redundant data in similar nodes, and the time complexity is $O(z \times K_1)$.

Hence, we can obtain that the complexity of the model scheme is $\max(O(f_1 \times n \times K), O(f_1 \times K \times q), O(f_2 \times z), O(f_2 \times K_1))$.

3.3.2. Energy Consumption. Most of the energy in the sensor node is consumed by its transceiver module. The channel model of the transmitter has two kinds of free space models and multipath fading models, and the energy consumption is related not only to the amount of data but also to the transmission distance d. Therefore, the energy consumption $E_{TX}(N, d)$ of the node to send N-bit data is

$$E_{TX}(N, d) = \begin{cases} N \times E_{\text{elec}} + N \times \varepsilon_{mp} \times d^4, d > d_0, \\ N \times E_{\text{elec}} + N \times \varepsilon_{fs} \times d^2, d \le d_0, \end{cases} \quad (17)$$

where E_{elec} represents the energy consumption of the circuit sending or receiving data and ε_{mp} and ε_{fs}, respectively, represent the energy consumption of the signal amplifier:

$$d_0 = \left(\frac{\varepsilon_{fs}}{\varepsilon_{mp}}\right)^{1/2}. \quad (18)$$

The energy consumption $E_{RX}(N)$ of the node receiving N-bit data is

$$E_{RX}(N) = N \times E_{\text{elec}}. \quad (19)$$

The energy consumption of nodes processing N-bit data is

$$E_P(N) = N \times E_P, \quad (20)$$

where E_P represents the energy consumption of processing unit data. The node's remaining energy consumption E_r is

$$E_r = E_0 - (E_{TX}(N_1, d) + E_{RX}(N_2) + E_P(N_3)), \quad (21)$$

where E_r represents the remaining energy consumption of the node, E_0 represents the initial energy, N_1 represents the total amount of data transmitted, N_2 represents the total amount of received data, and N_3 represents the total amount of data processing.

4. Experimental Results

4.1. Experimental Setup. To verify the effectiveness of the proposed method, the temperature sensing data from the Intel Berkeley Laboratory are used [35]; these data include 54 nodes, and each node collects sensing data every 0.5 minutes. The map is shown in Figure 4. To verify the deredundancy ratio of edge sensing data and the network lifetime, the data transmission model and node energy consumption model are adopted. The experiments consider the following metrics: the deredundancy ratio, the deredundancy error, the influence of the amount of similar data clusters K_1 on the deredundancy ratio, and the energy consumption. The proposed HMDA will be compared with the TCDA, TSDA, and Dat algorithms [9]. The parameters and their values are shown in Table 2.

4.2. Performance Evaluation. First, the performance results of the three stages are analyzed separately. In the first stage, clustering classification positions of similar nodes are obtained; in the second stage, clustering classification of similar data nodes is obtained; in the third stage, as a result of the first and second stages, the generated sensing data are made deredundant by means of random weighting. Second, the influence of K_1 on the deredundancy ratio in the second stage is analyzed. Finally, the energy consumption is analyzed with Dat [9], TCDA, TSDA, and HMDA.

In the first stage, the sink performs clustering according to the nodes' coordinate positions by running the improved k-means clustering algorithm. K is assumed to be 4, and $\beta = \{0, 0.3, 0.5, 0.7, 1\}$. The results of the four clusters also change significantly as β varies. The diamond in the figure represents the cluster center of the four clusters. The node's cluster distribution probability is shown in Table 3, and the clustering results are shown in Figure 5.

According to the probability ratio, the sink classifies the nodes that are prone to change into corresponding clusters. As shown in Table 3, the classification results are $C_1 = \{0, 2, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33\}$, $C_2 = \{1, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44\}$, $C_3 = \{3, 4, 5, 6, 7, 8, 9, 45, 46, 47, 48, 49, 50, 51, 52, 53\}$, and $C_4 = \{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$.

In the second stage, the CHs perform Gaussian mixture clustering. By successively acquiring edge sensing data from nodes within each cluster, the CHs can analyze data similarity according to further improve the deredundancy ratio. Similar classification results in cluster C_1 are shown in Figure 6.

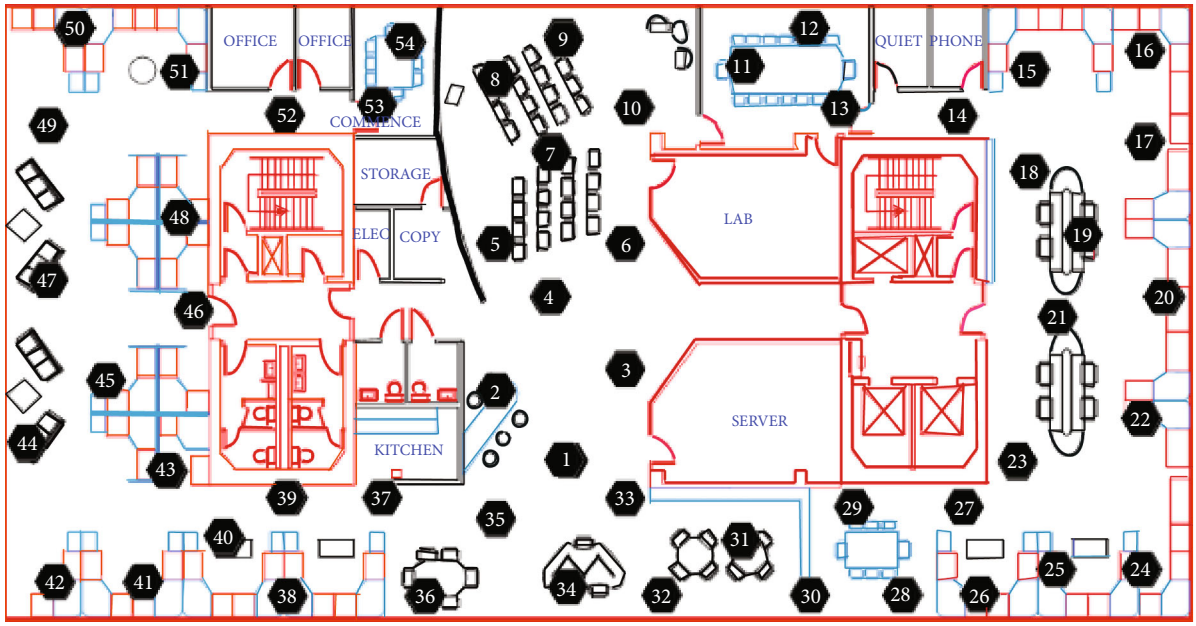


Figure 4: The map of sensor nodes in the experiment.

Table 2: Parameters of the experiments.

Parameter	Value
Duration	Feb. 28~Apr. 5, 2004
Area coverage	42 m × 33 m
Observation parameter	Temperature
Node size of WSNs	54
The number of perceived data collected/ten thousand	230
Collection interval (s)	31
Distance of CHs and sink (m)	10
E_{elec} (nJ/bit)	50
ε_{mp} (pJ/bit/m^4)	0.0013
ε_{fs} (pJ/bit/m^2)	100
E_P (nJ/bit)	5
E_0 (J)	5

Table 3: Node distribution probability in clusters.

Node ID	Cluster 1	Cluster 2	Cluster 3	Cluster 4
0	60%	40%	0	0
2	60%	40%	0	0
5	0	0	80%	20%
9	0	0	80%	20%
10	0	0	40%	60%
19	20%	0	0	80%
20	20%	0	0	80%
32	60%	40%	0	0
33	60%	40%	0	0
45	0	40%	60%	0
46	0	40%	60%	0

As shown in Figure 6, C_1 is divided into 4 subclusters: $C_{11} = \{22, 25, 28, 30, 32\}$, $C_{12} = \{23, 24, 26\}$, $C_{13} = \{27, 29, 31, 33\}$, and $C_{14} = \{0, 2, 21\}$. Similarly, cluster C_2 includes 3 subclusters: $C_{21} = \{1, 34, 35, 36\}$, $C_{22} = \{37, 38, 39\}$, and $C_{23} = \{40, 41, 43, 44\}$; C_3 is classified into $C_{31} = \{3, 4, 5, 6, 7\}$, $C_{32} = \{8, 9, 45, 46\}$, and $C_{33} = \{47, 48, 49, 50, 51, 52, 53\}$; and C_4 is divided into $C_{41} = \{10, 11, 12\}$, $C_{42} = \{13, 14, 15, 16\}$, and $C_{43} = \{17, 18, 19, 20\}$.

In the third stage, the data in similar clusters will be randomly weighted to optimize the redundancy ratio. For subcluster $C_{13} = \{27, 29, 31, 33\}$ in cluster C_1, the deredundancy performance, redundancy error, and mean square error are shown in Figures 7 and 8 and Table 4, respectively.

As shown in Figure 7, the sensing data of subcluster C_{13} tend to be the middle values with randomly weighted optimization. The sensing data of node 29 and node 33 are close to the deredundancy results. However, the values of node 27 and node 31 are relatively far away. From Figure 8, we can see that the mean square errors of nodes 29 and 33 are relatively lower than those of nodes 27 and 31. According to the results in Table 4, it can be seen that the mean square errors of nodes 27, 29, 31, and 33 are 0.035, 0.004, 0.034, and 0.006, respectively, which indicates that even if the data are similar, there are still differences between the sensing data. Therefore, for the methods of data similarity analysis with the coordinates of nodes, the lack of spatial correlation analysis can cause greater errors. Multistage clustering improves the accuracy of sensing data similarity.

Since the deredundancy ratio of MHCSD is related to the data similarity clustering K_1 in the second stage and the value

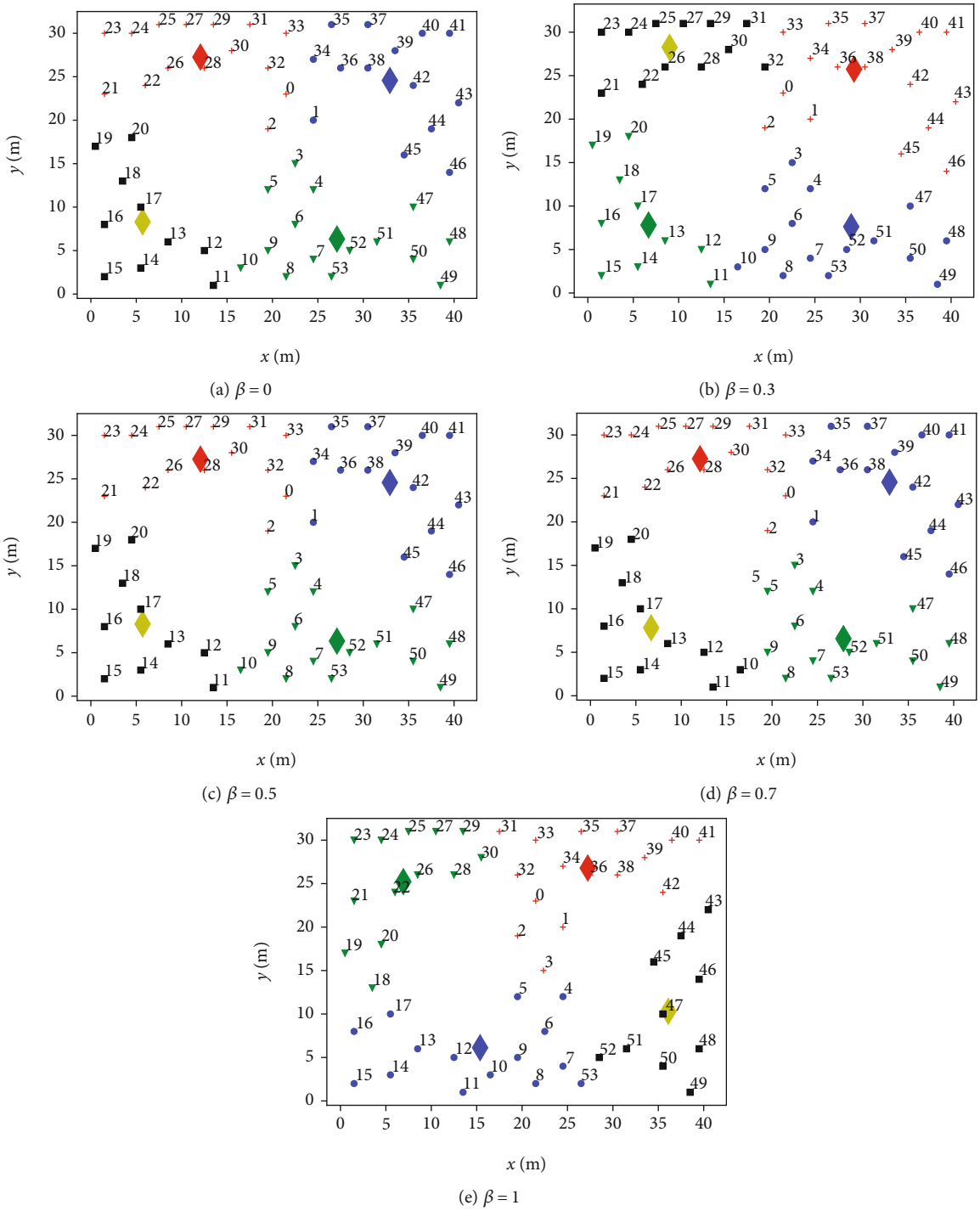


Figure 5: Node clustering of the improved k-means scheme with varying β.

of K_1 affects the accuracy of the data correlation, the effect of K_1 on the deredundancy ratio is shown in Figure 9.

As shown in Figure 9, the deredundancy ratio gradually decreases as K_1 increases. When $K_1 = 1$, it indicates that clusters C_1, C_2, C_3, and C_4 are not divided into any similar subclusters, and MHCSD treats all nodes in clusters C_1, C_2, C_3, and C_4 as redundant nodes, which perform random weighting to optimize sensing data. Therefore, the deredundancy ratio is maximized. However, when $K_1 = 1$, it is equivalent to clustering all nodes by position similarity without considering the data similarity cluster, which leads to a larger error. When $K_1 = 10$, MHCSD classifies the nodes into 10 similar subdata clusters in each cluster C_1, C_2, C_3, and C_4 and randomly weights them to deduplicate the sensing data. Hence, the redundancy ratio is the lowest, which guarantees the accuracy of the deredundant data. To ensure both the

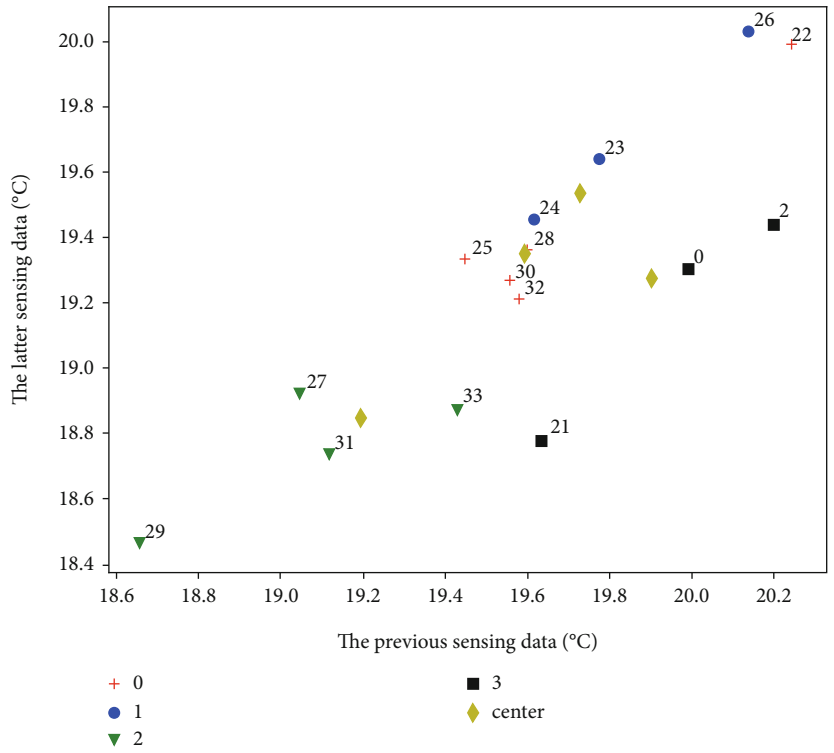


Figure 6: Data similarity distribution of C_1.

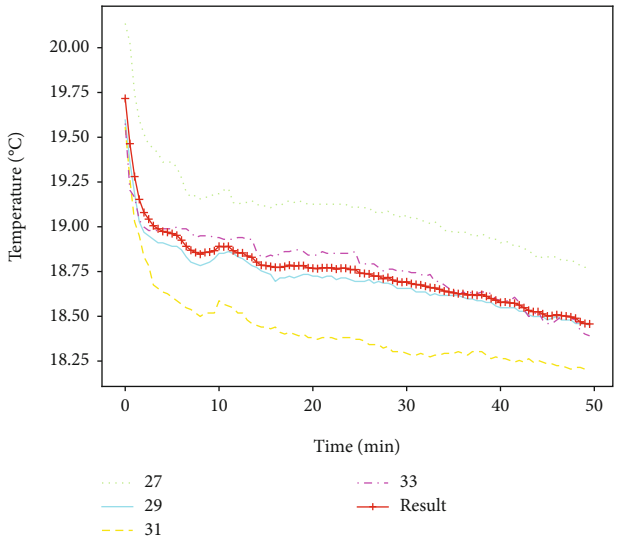


Figure 7: Deredundancy result of C_{13}.

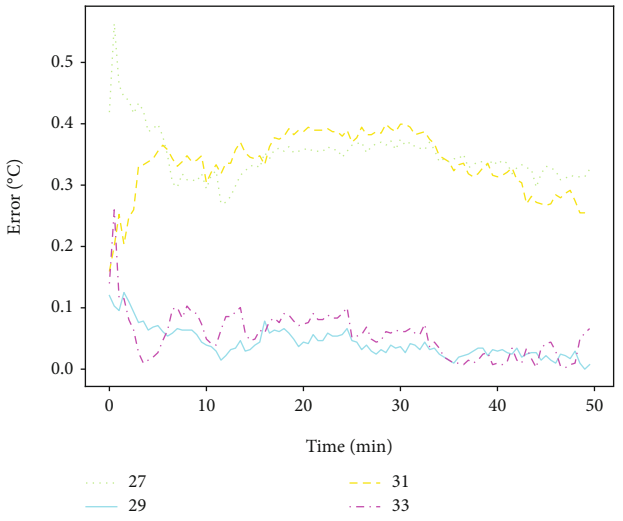


Figure 8: Deredundancy error of C_{13}.

accuracy of the data and the deredundancy ratio of the data, we set $K_1 = 4$ in the following performance analysis.

Figure 10 shows that as the number of nodes increases, the deredundancy ratio also increases and varies between 65% and 75%. When the number of nodes is 23, the deredundancy ratio is the highest (75%). When the number of nodes is less than 3, the proposed scheme omits the spatial correlation deredundancy and transmits the sensing data to the corresponding CHs, which degrades the deredundancy ratio.

Table 4: The mean square errors of nodes in C_{13}.

Node	Mean square error
27	0.035
29	0.004
31	0.034
33	0.006

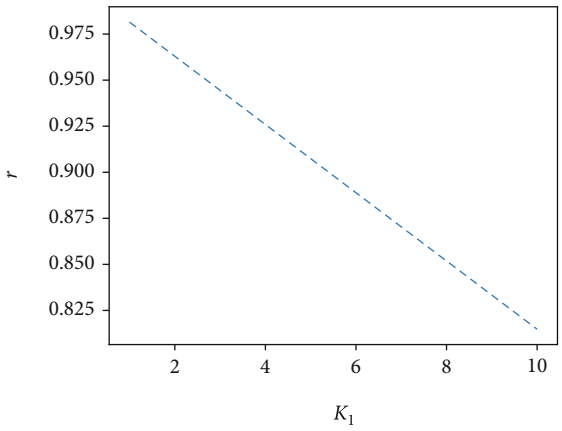


Figure 9: Effect of K_1 on deredundancy ratio.

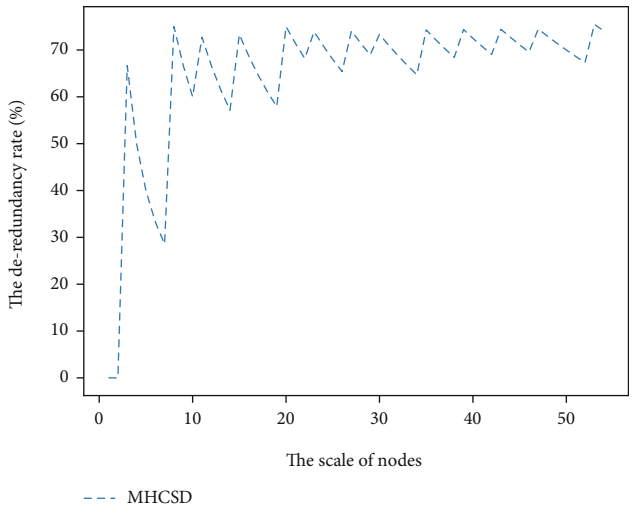


Figure 10: The deredundancy ratio of MHCSD.

When the number of nodes is larger than 3, MHCSD performs a spatial correlation deredundancy algorithm, deduplicates redundant nodes in clusters, and obviously improves the deredundancy ratio.

As seen from Figure 11, the deredundancy ratio of the HMDA algorithm varies between 97.50% and 98.0%, which is obviously higher than those of TCDA and Dat. Compared with TCDA and Dat, the deredundancy ratio of HMDA increases by 1.7% and 4.7%, respectively. Therefore, HMDA combines MHCSD and TCDA to reduce redundant data comprehensively and can further remove 70% of the redundant data. Additionally, the accuracy of the deredundancy nodes is maintained between 0.004 and 0.035, and within the allowable error range for a user, the deredundancy ratio reaches the highest. The results in Figure 11 also verify that HMDA is effective in improving the deredundancy ratio based on spatial-temporal correlations.

The energy consumed by different schemes is shown in Figure 12. The energy consumed by the four algorithms increases gradually as the number of nodes increases. Among the different schemes, the energy consumption of HMDA is much lower than those of the other three algorithms. For

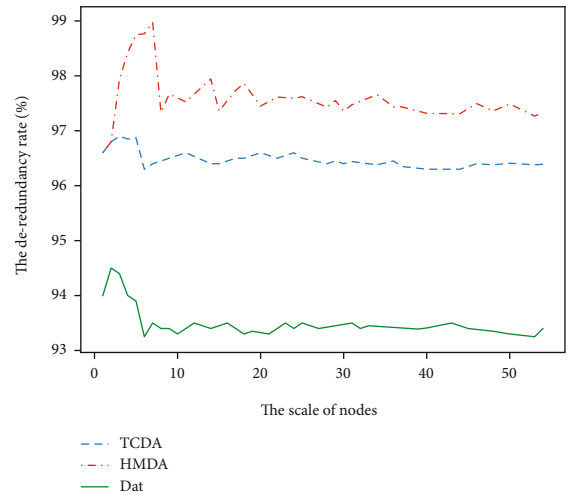


Figure 11: Deredundancy ratio of different schemes.

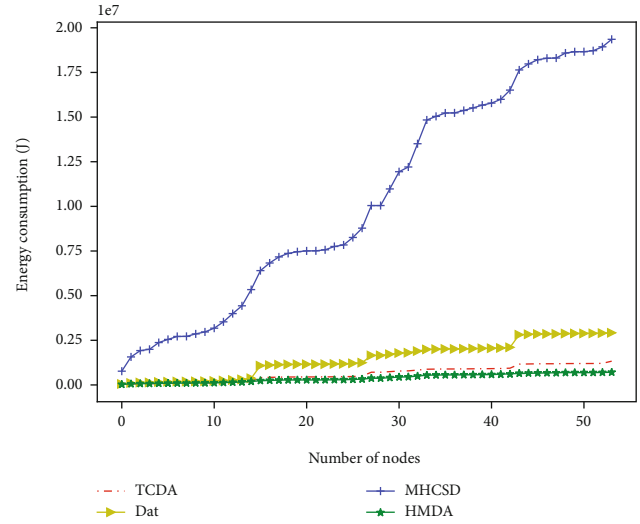


Figure 12: Energy consumptions of different schemes.

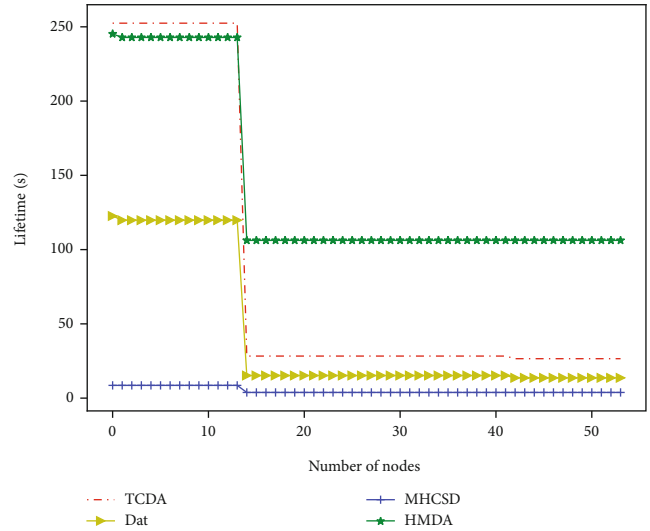


Figure 13: Lifetimes of different schemes.

the proposed HMDA scheme, energy consumption increases very slowly. When the number of nodes is 50, the energy consumption of HMDA is only 0.12 J, which is obviously lower than those of the other three algorithms. The reason is that HMDA will adaptively perform both spatial correlation and temporal correlation analyses.

The network lifetimes of different schemes are shown in Figure 13. When the number of nodes is lower than 15, the lifetimes of HMDA, TCDA, Dat, and MHCSD remain stable at 260 s, 250 s, 120 s, and 15 s, respectively. The reason is that the deredundancy ratios of the 4 schemes are 97.5%, 96.3%, 93%, and 70%, which ensures that all nodes perform the same data processing scheme with constant energy consumption. It is obvious that the lifetime of HMDA is longer than that of the other 3 schemes. Especially when the number of nodes increases to 50, the lifetime of HMDA is 109 s, which is 12.6, 3.0, and 3.9 times higher than those of MHCSD, TCDA, and Dat, respectively. The results in Figures 11–13 demonstrate that the proposed HMDA scheme can achieve better performance in terms of the deredundant ratio, energy consumption, and network lifetime.

5. Conclusion

Focusing on the problem of data redundancy in WSNs, a multistage hierarchical clustering deredundancy algorithm is proposed to decrease the additional power consumption and extend the lifetime of a WSN. Based on the improved k-means clustering method, all nodes are classified according to the node position information and temporal similarity. The Gaussian hybrid clustering method is adopted to improve the redundant similarity of edge nodes. According to the secondary classification results, the sensing data generated by the redundant nodes are randomly weighted to remove the redundant data. Detailed analysis and experimental results show that, compared with the existing schemes, the proposed scheme is superior in terms of the deredundancy ratio, power consumption, and lifetime of a WSN.

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 are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Science Foundation of China (No. 61772562, 62062019), the Key Project of Hubei Provincial Science and Technology Innovation Foundation of China (No. 2018ABB1485), the Hubei Provincial Natural Science Foundation of China (No. 2019CFB815), the Fundamental Research Funds for the Central Universities (No. CZP19004), and the Youth Elite Project of State Ethnic Affairs Commission of China (No. 2016-3-08).

References

[1] P. Zeng, B. Pan, K.-K. R. Choo, and H. Liu, "MMDA: multidimensional and multidirectional data aggregation for edge computing-enhanced IoT," *Journal of Systems Architecture*, vol. 106, article 101713, 2020.

[2] 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.

[3] 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.

[4] 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.

[5] W. Tang, X. Zhao, W. Rafique, L. Qi, W. Dou, and Q. Ni, "An offloading method using decentralized P2P-enabled mobile edge servers in edge computing," *Journal of Systems Architecture*, vol. 94, pp. 1–13, 2019.

[6] S. Wan, Z. Gu, and Q. Ni, "Cognitive computing and wireless communications on the edge for healthcare service robots," *Computer Communications*, vol. 149, pp. 99–106, 2020.

[7] G. B. Tayeh, A. Makhoul, C. Perera, and J. Demerjian, "A spatial-temporal correlation approach for data reduction in cluster-based sensor networks," *IEEE Access*, vol. 7, pp. 50669–50680, 2019.

[8] A. K. Idrees, A. K. M. Al-Qurabat, C. A. Jaoude, and W. L. Al-Yaseen, "Integrated divide and conquer with enhanced k-means technique for energy-saving data aggregation in wireless sensor networks," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 973–978, Tangier, Morocco, June 2019.

[9] R. Alhussaini, A. K. Idrees, and M. A. Salman, "Data transmission protocol for reducing the energy consumption in wireless sensor networks," in *International Conference on New Trends in Information and Communications Technology Applications*, pp. 35–49, Baghdad, Iraq, 2018.

[10] S. Chowdhury, A. Roy, A. Benslimane, and C. Giri, "On semantic clustering and adaptive robust regression based energy-aware communication with true outliers detection in WSN," *Ad Hoc Networks*, vol. 94, article 101934, 2019.

[11] Y. Fathy, P. Barnaghi, and R. Tafazolli, "An adaptive method for data reduction in the internet of things," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 729–735, Singapore, February 2018.

[12] W. M. Elsayed, H. M. El-Bakry, and S. M. EL-Sayed, "Data reduction using integrated adaptive filters for energy-efficient in the clusters of wireless sensor networks," *IEEE Embedded Systems Letters*, vol. 11, no. 4, pp. 119–122, 2019.

[13] N. A. M. Alduais, J. Abdullah, A. Jamil, and H. Heidari, "Performance evaluation of real-time multivariate data reduction models for adaptive-threshold in wireless sensor networks," *IEEE Sensors Letters*, vol. 1, no. 6, pp. 1–4, 2017.

[14] L. Tan and M. Wu, "Data reduction in wireless sensor networks: a hierarchical LMS prediction approach," *IEEE Sensors Journal*, vol. 16, no. 6, pp. 1708–1715, 2016.

[15] N. R. Roy and P. Chandra, "Analysis of data aggregation techniques in WSN," *International Conference on Innovative Computing and Communications*, vol. 2, pp. 571–581, 2020.

[16] D. Yuvaraj, M. Sivaram, and S. Navaneetha Krishnan, "Intelligent detection of untrusted data transmission to optimize energy in sensor networks," *Journal of Information and Optimization Sciences*, vol. 41, no. 3, pp. 799–811, 2020.

[17] R. Banerjee, S. Chatterjee, and S. Das Bit, "Performance of a partial discrete wavelet transform based path merging compression technique for wireless multimedia sensor networks," *Wireless Personal Communications*, vol. 104, no. 1, pp. 57–71, 2019.

[18] S. Diwakaran, B. Perumal, and K. Vimala Devi, "A cluster prediction model-based data collection for energy efficient wireless sensor network," *The Journal of Supercomputing*, vol. 75, no. 6, pp. 3302–3316, 2019.

[19] C. Xu, K. Wang, Y. Sun, S. Guo, and A. Y. Zomaya, "Redundancy avoidance for big data in data centers: a conventional neural network approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 104–114, 2020.

[20] S. Kumar and V. K. Chaurasiya, "A strategy for elimination of data redundancy in Internet of things (IoT) based wireless sensor network (WSN)," *IEEE Systems Journal*, vol. 13, no. 2, pp. 1650–1657, 2019.

[21] J. Zhang, H. Huang, Y. Luo, Y. Fan, and G. Yang, "Immunization-based redundancy elimination in mobile opportunistic networks- generated big data," *Future Generation Computer Systems*, vol. 79, pp. 920–927, 2018.

[22] Y. Li, Z. Lv, J. Zhao, and Z. Pan, "Improving performance of medical image fusion using histogram, dictionary learning and sparse representation," *Multimedia Tools and Applications*, vol. 78, no. 24, pp. 34459–34482, 2019.

[23] A. G. Soundari and V. L. Jyothi, "Energy efficient machine learning technique for smart data collection in wireless sensor networks," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 1089–1122, 2020.

[24] M. Rida, A. Makhoul, H. Harb, D. Laiymani, and M. Barhamgi, "EK-means: a new clustering approach for datasets classification in sensor networks," *Ad Hoc Networks*, vol. 84, pp. 158–169, 2019.

[25] S. Y. Lin, J. B. Li, and C. T. Yu, "Dynamic data driven-based automatic clustering and semantic annotation for internet of things sensor data," *Sensors and Materials*, vol. 31, no. 6, pp. 1789–1801, 2019.

[26] M. K. Alam, A. A. Aziz, S. A. Latif, and A. Awang, "Data clustering technique for in-network data reduction in wireless sensor network," in *2019 IEEE Student Conference on Research and Development (SCOReD)*, Bandar Seri Iskandar, Malaysia, October 2019.

[27] D. Ruby and J. Jeyachidra, "Semaphore based data aggregation and similarity findings for underwater wireless sensor networks," *International Journal of Grid and High Performance Computing*, vol. 11, no. 3, pp. 59–76, 2019.

[28] R. Wan, N. Xiong, Q. Hu, H. Wang, and J. Shang, "Similarity-aware data aggregation using fuzzy c-means approach for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, Article ID 59, 2019.

[29] H. Harb and A. Makhoul, "Energy-efficient scheduling strategies for minimizing big data collection in cluster-based sensor networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 3, pp. 620–634, 2019.

[30] R. Nawaz Jadoon, W. Y. Zhou, I. A. Khan, M. A. Khan, and W. Jadoon, "EEHRT: energy efficient technique for handling redundant traffic in zone-based routing for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 7502140, 12 pages, 2019.

[31] L. Yang, H. Zhu, H. Wang, K. Kang, and H. Qian, "Data censoring with network lifetime constraint in wireless sensor networks," *Digital Signal Processing*, vol. 92, pp. 73–81, 2019.

[32] A. Jarwan, A. Sabbah, and M. Ibnkahla, "Data transmission reduction schemes in WSNs for efficient IoT systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1307–1324, 2019.

[33] J. Tan, W. Liu, M. Xie et al., "A low redundancy data collection scheme to maximize lifetime using matrix completion technique," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, Article ID 5, 2019.

[34] K. K. Lim, J. S. Park, and J. G. Shon, "Differential data processing technique to improve the performance of wireless sensor networks," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4489–4504, 2019.

[35] S. Madden, "Intel Berkeley Research Lab," 2004, http://db.csail.mit.edu/labdata/labdata.html.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6621451, 14 pages
https://doi.org/10.1155/2021/6621451

Research Article

An Online Semisupervised Learning Model for Pedestrians' Crossing Intention Recognition of Connected Autonomous Vehicle Based on Mobile Edge Computing Applications

Shicai Ji,[1] Ying Peng,[1] Hongjia Zhang,[2] and Shengbo Wu[2]

[1]*School of Vehicle Engineering, Shandong Transport Vocational College, Weifang, Shandong 261206, China*
[2]*School of Automobile, Chang'an University, Xi'an, Shaanxi 710064, China*

Correspondence should be addressed to Hongjia Zhang; zhanghongjia@chd.edu.cn

Received 25 November 2020; Revised 7 January 2021; Accepted 19 January 2021; Published 5 February 2021

Academic Editor: Shaohua Wan

One of the major challenges that connected autonomous vehicles (CAVs) are facing today is driving in urban environments. To achieve this goal, CAVs need to have the ability to understand the crossing intention of pedestrians. However, for autonomous vehicles, it is quite challenging to understand pedestrians' crossing intentions. Because the pedestrian is a very complex individual, their intention to cross the street is affected by the weather, the surrounding traffic environment, and even his own emotions. If the established street crossing intention recognition model cannot be updated in real time according to the diversity of samples, the efficiency of human-machine interaction and the interaction safety will be greatly affected. Based on the above problems, this paper established a pedestrian crossing intention model based on the online semisupervised support vector machine algorithm (OS^{3}VM). In order to verify the effectiveness of the model, this paper collects a large amount of pedestrian crossing data and vehicle movement data based on laser scanner, and determines the main feature components of the model input through feature extraction and principal component analysis (PCA). The comparison results of recognition accuracy of SVM, S^{3}VM, and OS^{3}VM indicate that the proposed OS^{3}VM model exhibits a better ability to recognize pedestrian crossing intentions than the SVM and S^{3}VM models, and the accuracy achieves 94.83%. Therefore, the OS^{3}VM model can reduce the number of labeled samples for training the classifier and improve the recognition accuracy.

1. Introduction

According to the annual road traffic accident statistics report released by the Traffic Administration Bureau of the Ministry of Public Security of the People's Republic of China, the number of people who died in traffic accidents in China in 2019 was 62,763 and the number of people injured was 256,101. Among them, accidents between pedestrians and vehicles resulted in 17,473 deaths and 45,495 injuries [1].

Pedestrians are vulnerable road users and require active protection. Both autonomous driving and connected cars are designed to provide greater safety benefits [2, 3]. Autonomous vehicles need to have the ability to determine the intentions of other road users and communicate with them. This interaction is crucial between vehicles and pedestrians. However, pedestrian crossing intention is a relatively complex process, which depends on various factors, such as pedestrian status and traffic environment [4].

According to a recent report by Google's autonomous vehicles, 90% of the failures of autonomous vehicles occur on busy streets, and 10% of them are due to misrecognition of pedestrian intention. Among the various behaviors of pedestrians, street crossing is the most important one, which is related to the safety of pedestrians. Through visual communication, gesture communication, and even auditory communication with pedestrians, human drivers can easily

recognize the pedestrian's crossing intention. However, for autonomous vehicles, such communication with pedestrians is quite challenging [5, 6].

Most of the current researches mainly reveal pedestrians' crossing intentions from pedestrian movement data and pedestrian posture data. Pedestrian's intention to cross the street is actually a classification problem of behavior sequence data. There is a strong front and back dependency between sequence data [3].

Schulz and Stiefelhagen [7] proposed a pedestrian crossing intention recognition method combining multiple interactive multiple model filters and latent-dynamic conditional random field model. The input parameters are mainly position and speed. Varytimidis et al. [8] used the convolutional neural network (CNN) algorithm to extract the features of the pedestrian's posture and then recognized the pedestrian's crossing intention based on the pedestrian's head posture. The algorithms used are the support vector machine (SVM) and artificial neural network (ANN). Völz et al. [9] used SVM, CNN, and long- and short-term memory networks (LSTM) to identify pedestrian crossing intentions. The input feature parameters are mainly the distance between the pedestrian and the zebra crossing, and the distance between the vehicle and the zebra crossing. Park and Lee [10] obtained the EMG signal during pedestrian movement and used CNN for learning. It was found that the features can effectively decode the pedestrian's movement intention. Zhang et al. [11] proposed a pedestrian crossing intention recognition model based on the attention mechanism of long- and short-term memory networks (AT-BiLSTM). The input feature parameters are mainly pedestrian speed, distance between pedestrian and vehicle, and distance between vehicle and zebra crossing. Schneemann and Heinemann [12] proposed a pedestrian crossing intention model based on contextual features, using a support vector machine algorithm. The descriptor captured the movement of pedestrians relative to the road and the spatial layout of other scene elements in a generic manner. It showed that context-based data are good indicators for crossing prediction. Zhao et al. [13] proposed a pedestrian crossing intention model based on improved naive Bayesian networks. The input feature data source is Lidar. Camara et al. proposed an intention heuristic model. The input feature parameters include pedestrian trajectory, vehicle trajectory, and relative position. It was found that the model recognition accuracy was high, reaching 96%. Fang et al. [14] established a pedestrian crossing intention model based on SVM and found that the model can accurately identify pedestrian crossing intentions with an accuracy rate of 93%. The input of the model is the feature parameters of pedestrian body posture. Škovierová et al. [15] collected the position, speed, and orientation information of all traffic participants of pedestrians, and realized the recognition of pedestrian intention through the Bayesian network. Quintero et al. [16, 17] used pedestrian posture features to recognize pedestrian crossing intentions and retrogrades. Although the recognition accuracy is high, the recognition time lags to a certain extent.

Through the above review, it can be seen that the current pedestrian crossing intention recognition is mainly based on data-driven and pedestrian posture feature-driven, and recognition algorithms are traditional supervised learning algorithms. However, the shortcomings of the current pedestrian street crossing intention model are also obvious. First of all, some pedestrian street crossing intention models are established based on pedestrian posture data. When the pedestrian's head is blocked or the sunlight is too strong, it will seriously affect the recognition accuracy of the intention model. Secondly, the current pedestrian street crossing intention model is mainly based on supervised learning. A large number of data labels need to be manually labeled. In the era of big data, this method is very time-consuming. Finally, the current pedestrian street crossing intention model cannot perform online self-learning. When faced with some special situations, it cannot perform self-learning based on data, which greatly affects the generalization performance of the model.

Samples are known to have certain features as a training data set, build a model, and then use this model to classify unknown samples. This method is called supervised learning and is the most commonly used machine learning method. Generally speaking, each set of feature data corresponds to a specific label when the classification model is trained [18, 19]. Semisupervised learning (SSL) [20] is a key issue in the field of pattern recognition and machine learning. It is a learning method that combines supervised learning and unsupervised learning. Semisupervised learning uses a large amount of unlabeled data and simultaneously uses labeled data for pattern recognition. When using semisupervised learning, it will require as few people as possible to do the work, and at the same time, it can bring relatively high accuracy. Online learning can quickly make model adjustments based on online feedback data and improve the accuracy of online predictions. The process of online learning includes presenting the prediction results of the model to the user and then collecting the user's feedback data, which is then used to train the model to form a closed-loop system [21, 22]. Online semisupervised learning is the product of the fusion of semisupervised learning and online learning. While the labeled and unlabeled samples can be stored, it also has the characteristics of online learning. The online semisupervised learning algorithm is a sequence of continuous learning cycles ongoing. In each learning cycle, the learner is given a training sample and is required to predict the label of the sample in the case of training unlabeled samples [23, 24].

A large number of sensors, cameras, millimetre wave radars, and Lidar devices are installed on the connected autonomous vehicles (CAVs). These devices will generate abundant data, which will lead to the lack of storage resources, computing resources, and communication resources. Due to the limitations of the equipment in computing power and storage performance, the CAVs cannot perform computationally intensive tasks. At this point, the task needs to be sent to the server, which will process the task, and then the processing results will be fed back to the vehicle. The data processing based on the traditional cloud computing model will not only lead to long task execution delays but also increase the energy consumption. Centralized cloud servers are far away from terminal devices, which leads to inefficiency in computing-

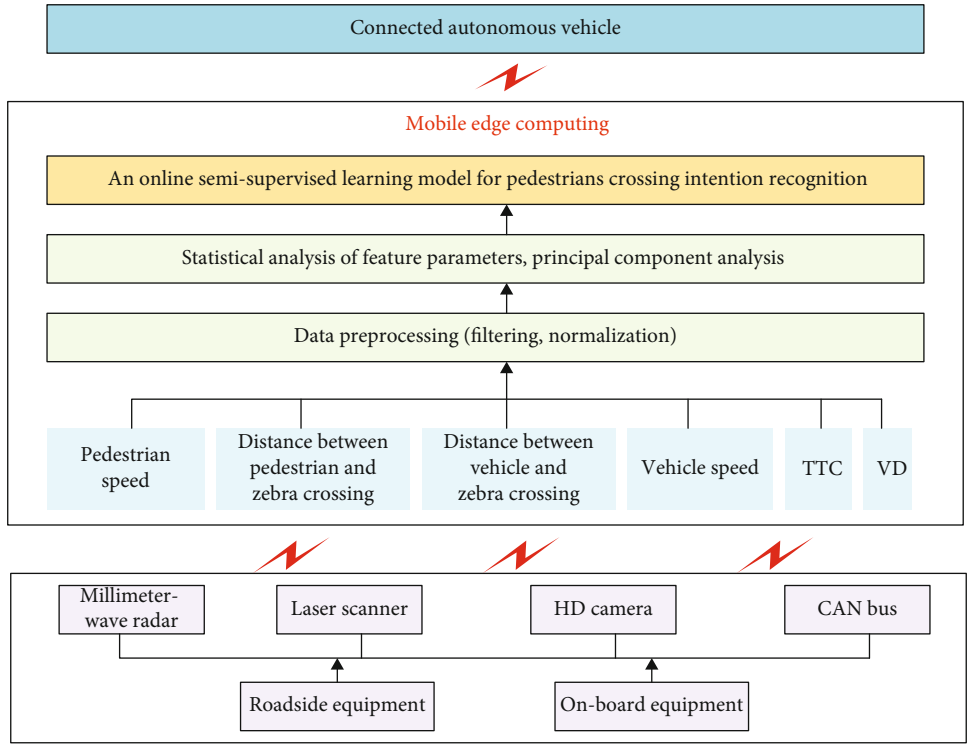


Figure 1: The structure of pedestrian intention recognition based on mobile edge computing.

intensive environments. At the same time, the transmission of computing resources to the cloud consumes energy, which may also reduce the service life of mobile batteries. In addition, the cloud computing patterns make it difficult to provide mobile users with complex memory utilization applications and higher data storage capacity. Mobile edge computing (MEC) is regarded as an effective way to solve the above problems. By deploying computing resources at the edge of the network, the delay-sensitive tasks such as collision prediction, surrounding vehicle and pedestrian intention prediction, vehicle avoidance control, and other tasks are assigned to the edge server for calculation, which can greatly reduce the communication delay and also can effectively improve the data security.

In this study, an online semisupervised learning model for pedestrians' crossing intention recognition based on mobile edge computing technology was established. The MEC technology is an ideal choice for CAVs, which can play a key role in assisting the intelligent vehicles. Therefore, edge intelligence was employed to acquire and process pedestrian and vehicle data at the edge of the network, and the pedestrian intention recognition result was fed back to the decision-making system of CAVs in time. By comparing the characteristics of supervised learning, semisupervised learning, online learning, and online semisupervised learning, we chose an online semisupervised learning algorithm to identify pedestrian crossing intentions. Existing supervised learning algorithms need to manually label data, and the model cannot be updated in real time. In view of this, this work proposed an online semisupervised support vector machine algorithm (OS^{3}VM). Based on the semisupervised support vector machine classification model, the local concave-convex process optimization (LCCCP) algorithm was employed to remark the soft labels of unmarked samples in an iterative way. Then, the greedy promotion algorithm was used to further update the dual variables to realize the online learning process of the S^{3}VM model. The proposed pedestrians' crossing intention recognition model has the structure shown in Figure 1.

2. Experimental and Data Collection

2.1. Experiment Site and Equipment. The experimental road is two-way four-lane, and the length of the zebra crossing is 12 m. The center of the road is separated by a double yellow line, and there is no fence or refuge island in the middle. The experimental site is a section without signal light control. The experimental site is relatively common in China and has a certain representativeness. The transportation elements are mainly cars and buses. Traffic flow is about 450 veh/h. Figure 2 shows the experiment site.

The main experimental equipment is a 4-layer laser scanner and high-definition (HD) camera. The laser scanner model was an IBEO LUX with a scanning frequency of 12.5 Hz, a detectable range of 0.3–200 m, and a vertical viewing angle of 3.2°FOV. It was mainly used to collect some objective parameters such as the vehicle speeds, crossing speeds of the pedestrians, and the distances between vehicles and pedestrians. The video capture equipment used a mini HD monitor with a video resolution of 1920 × 1080, which ensured the

Figure 2: Experiment site.

definition of the video and met the experimental requirement. HD cameras are mainly used to determine the age, gender, and group attributes of pedestrians. In order to avoid the observer effect, the equipment is installed 15 m away from the zebra crossing. Figure 3 shows the experiment equipment.

2.2. Intention Feature Extraction. In this work, pedestrians' crossing intentions are mainly divided into two categories, namely, "stopping" and "crossing". When the pedestrian's crossing intention is "stopping", it means that the pedestrian's speed at a certain distance from the zebra crossing is relatively large, and when the pedestrian reaches the curb, the pedestrian's crossing speed is 0. When the pedestrian's crossing intention is "crossing", it means that the pedestrian crosses the zebra crossing at the original speed, and the speed before crossing the street is not much different from the speed when crossing the street.

All experimental data were collected in sunny weather. Avoid the weather's interference with pedestrians' intention to cross the street. The experiment was carried out for about one month, and mainly collected the movement data of pedestrians and vehicles before crossing the street. Through data extraction, the pedestrian crossing intention feature parameters selected in this work include the pedestrian speed before crossing the street (PS), the distance between pedestrian and zebra crossing (DPZ), the distance between vehicle and zebra crossing (DVZ), vehicle speed (VS), time to collision (TTC), and vehicle deceleration (VD). In addition, the age, gender, and group attributes of pedestrians have a greater impact on pedestrians' intention to cross the street. There, it is also considered in this work. The detailed analysis of feature parameters is described as follows. This study focused on two pedestrian street crossing decisions: crossing and stopping. Two different locations were selected for the experiment. The experiment period is 22 days. We collected 900 samples of pedestrians who intended to stop and 900 samples of pedestrians who intended to cross. Pedestrian crossing intention recognition is actually a kind of sequence data recognition. In this work, we intercepted the data of pedestrians and vehicles 2 s before crossing the street for analysis. In other words, the input length of the feature parameters is 2 s.

The data was preprocessed before statistical analysis, mainly including data filtering and data normalization. Pedestrian speed and vehicle speed collected by laser scanner may have a step phenomenon in a short time. In order to minimize this phenomenon, the paper used Gaussian smoothing filter for data processing. In addition, since the vehicle speed and the distance between the vehicle and the zebra crossing are relatively large, in order to improve the recognition accuracy and training speed of the model, the data was normalized.

2.2.1. PS. When the pedestrian's crossing intention is "stopping", the mean PS before crossing the street is 2.48 km/h. When the pedestrian's crossing intention is "crossing", the mean PS before crossing the street is 4.15 km/h. As shown in Figure 4. The one-way analysis of variance (ANOVA) test found that the mean PS before crossing the street under the two kinds of crossing intentions was significantly different ($F(1.1776) = 375.31$, $p < 0.05$).

2.2.2. DPZ. When the pedestrian's crossing intention is "stopping", the mean DPZ before crossing the street is 0.63 m. When the pedestrian's crossing intention is "crossing", the mean DPZ before crossing the street is 0.99 m. As shown in Figure 5, the one-way ANOVA test found that the mean DPZ before crossing the street under the two kinds of crossing intentions was significant difference ($F(1.1776) = 160.64$, $p < 0.05$).

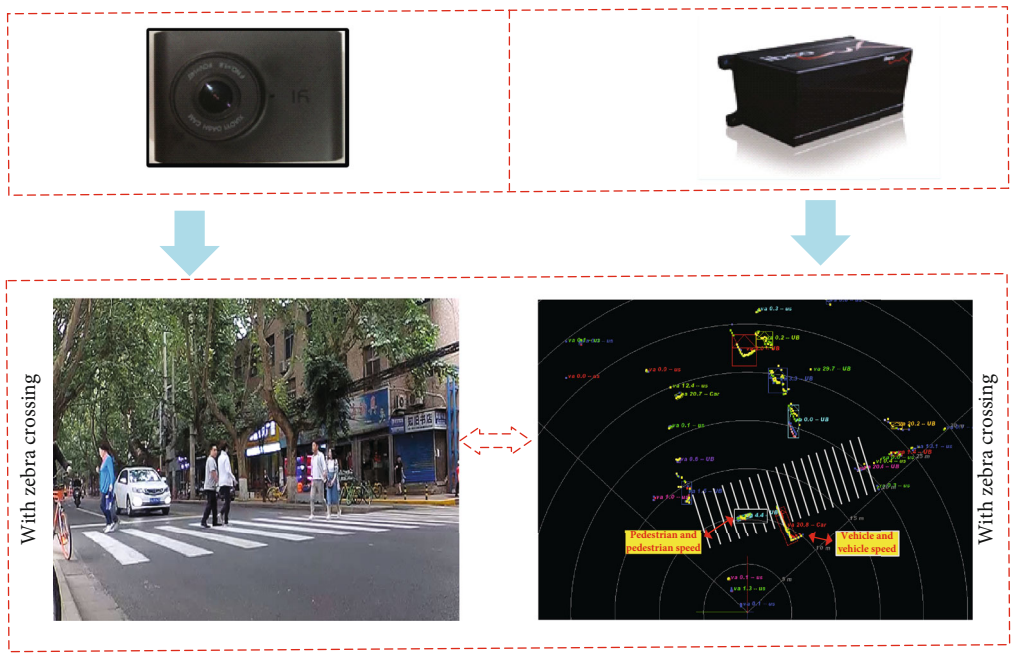


Figure 3: Experiment equipment.

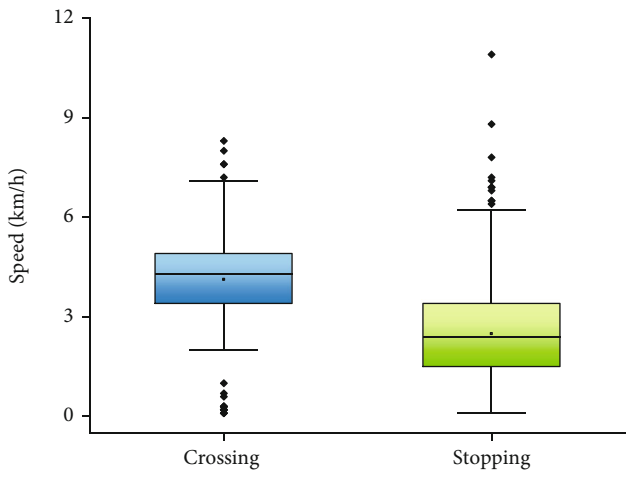


Figure 4: Comparison of PS in the intention of "crossing" and "stopping".

2.2.3. DVZ. When the pedestrian's crossing intention is "stopping", the mean DVZ before reaching the street is 19.59 m. When the pedestrian's crossing intention is "crossing", the mean DVZ before reaching the street is 38.56 m. As shown in Figure 6, the one-way ANOVA test found that the mean DVZ before crossing the street under the two kinds of crossing intentions was significant difference ($F(1.1776) = 660.64$, $p < 0.05$).

2.2.4. VS. When the pedestrian's crossing intention is "stopping", the mean VS before reaching the street is 28.91 km/h. When the pedestrian's crossing intention is "crossing", the mean VS before reaching the street is 28.90 km/h. As shown in Figure 7, the one-way ANOVA test found that the mean VS before crossing the street under the two kinds of crossing intentions was no significant difference ($F(1.1776) = 0.96$, $p > 0.05$).

2.2.5. TTC. When the pedestrian's crossing intention is "stopping", the mean TTC before arriving the street is 2.45 km/h. When the pedestrian's crossing intention is "crossing", the mean TTC before arriving the street is 4.50 s. As shown in Figure 8, the one-way ANOVA test found that the mean TTC before crossing the street under the two kinds of crossing intentions was no significant difference ($F(1.1776) = 410.45$, $p < 0.05$).

2.2.6. VD. When the pedestrian's crossing intention is "stopping", the mean VD before arriving the street is 2.26 m/s^2. When the pedestrian's crossing intention is "crossing", the mean VD before arriving the street is 1.20 m/s^2. As shown in Figure 9, the one-way ANOVA test found that the mean VD before crossing the street under the two kinds of crossing intentions was no significant difference ($F(1.1776) = 406.43$, $p < 0.05$).

2.2.7. Age, Gender, and Group. It is well known that the age, gender, and group attributes of pedestrians can significantly affect pedestrians' intention to cross the street. Middle-aged pedestrians tend to take risks, while the elderly is relatively conservative. Compared with male pedestrians, female pedestrians are also relatively conservative. In addition, group pedestrians are more radical than single pedestrians [25–27]. In order to improve the recognition accuracy of pedestrian crossing intention recognition model, we take pedestrian age and pedestrian gender as input variables to train the model. The pedestrians' age was divided according

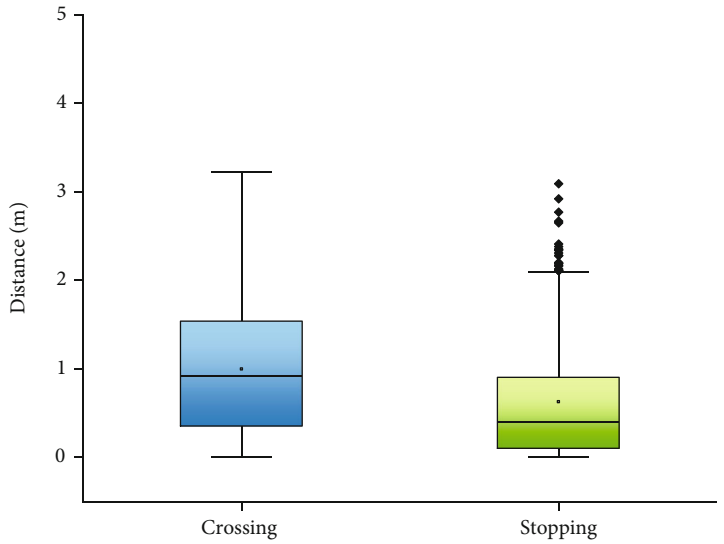


Figure 5: Comparison of DPZ in the intention of "crossing" and "stopping".

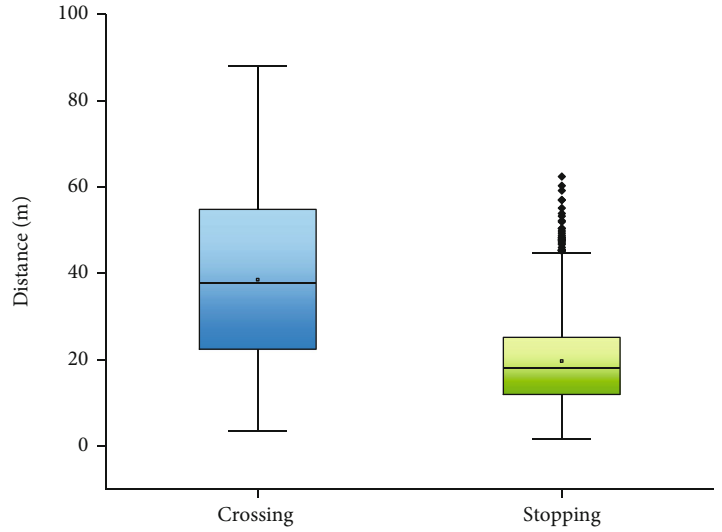


Figure 6: Comparison of DVZ in the intention of "crossing" and "stopping".

to natural observation, using the classification method mentioned in the references which define 18–30 as a youth, 30–59 as middle age, and >60 as old age [28, 29]. Hashimoto et al. [30] found that individuals or groups have great differences in pedestrian crossing behavior and use this attribute as input variable to train the intention recognition model.

2.3. Principal Component Analysis (PCA). Through feature analysis, it can be seen that only the vehicle speed of the above six feature parameters has nothing to do with pedestrian crossing intention. Although multiple features contain rich information, omission of features can be avoided. However, the long input feature parameters will slow down the recognition speed and reduce the recognition accuracy. In addition, the model may also be overfitting. Therefore, this paper used PCA to reduce the dimension of the feature parameters. On the basis of retaining the original feature parameter information, reduce the dimension of the parameters [31]. The PCA algorithm is used to reduce the dimensionality of five feature parameters of crossing intention, and the correlation between the variables and the principal components is shown in Table 1.

From Table 1, we can see that VD, PS, and PC1 have a strong correlation. TTC, DVZ, and PC2 have a strong correlation. TTC, DVZ, VD, and PC3 have a strong correlation. Figure 2 shows PCA feature extraction. It can be seen from Table 2 that the eigenvalues of the first three principal components are all greater than 1, so the first three principal components are selected to replace the original variables. The corresponding cumulative variance contribution rate is

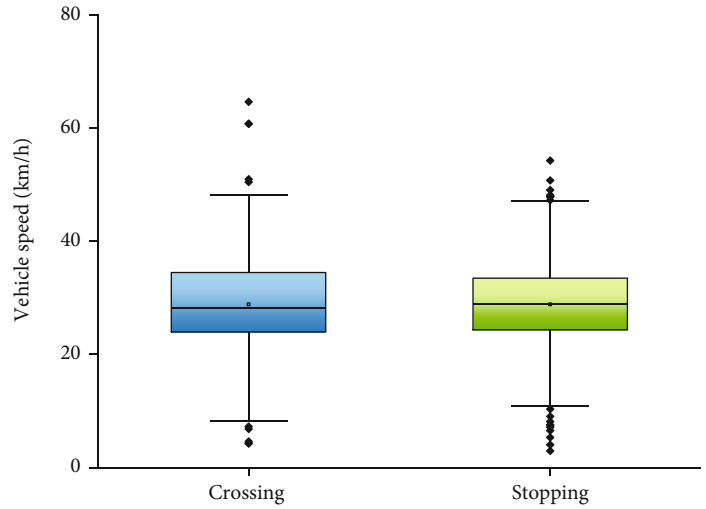


Figure 7: Comparison of VS in the intention of "crossing" and "stopping".

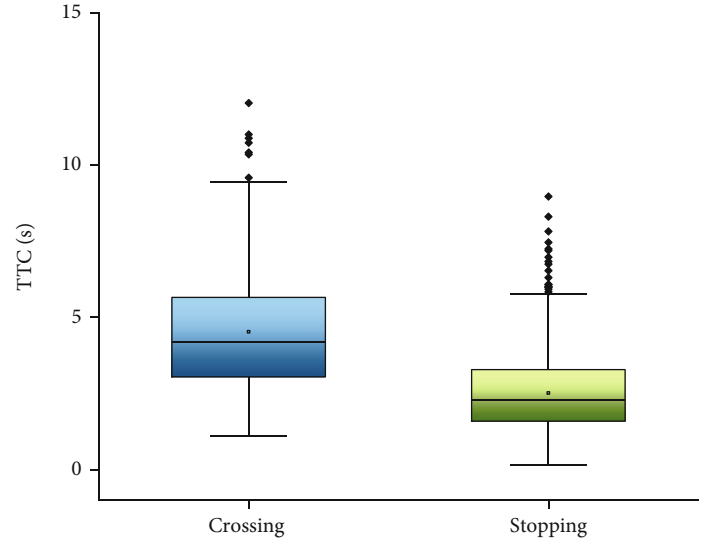


Figure 8: Comparison of TTC in the intention of "crossing" and "stopping".

91.92%, which shows that extracting the first 3 principal components to replace the original variables only loses 8.08% of the information. Therefore, this experiment selected the first three principal components (PC1~PC3) as the feature input of the pedestrian crossing intention model. In addition, the input feature parameters of the intention model also include pedestrian age, gender, and group attributes.

2.4. Sample Label. The experiment collected a total of 1800 data sets. *K*-means clustering is widely used in pattern recognition and sample labeling [32, 33]. In addition, *K*-means clustering also plays an important role in semisupervised learning. In this work, the labeled samples are divided into two categories according to certain characteristics through *K*-means clustering. When unlabeled samples enter the model, they are first clustered and labeled, and then further trained by a semisupervised learning algorithm. In this work, the intention of pedestrians to cross the street is divided into two categories, namely, "crossing" and "stopping".

3. Recognition Model Design

The online semisupervised learning (OSSL) algorithm possesses the advantages of both semisupervised learning and online learning algorithms [34, 35]. The semisupervised data stream containing marked samples and unmarked samples can be learned online simultaneously, and then, the recognition model can be updated in real time. In this paper, an online semisupervised support vector machine (OS^3VM) algorithm was established to identify the pedestrian crossing intentions. Based on the semisupervised support vector machine classification model, the local concave-convex process optimization (LCCCP) algorithm was employed to remark the soft labels of unmarked samples in an iterative

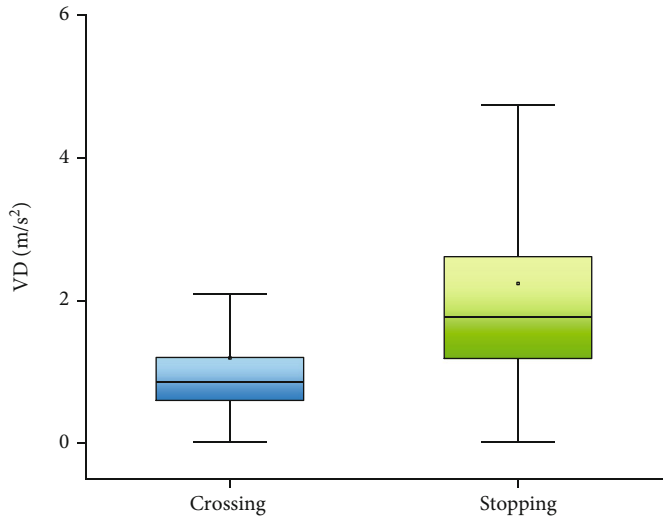


Figure 9: Comparison of VD in the intention of "crossing" and "stopping".

Table 1: Correlation between the variables and the principal components.

Variable	PC1	PC2	PC3	PC4	PC5
TTC	-0.11	0.79	0.45	-0.07	-0.02
DVZ	0.02	-0.69	0.67	-0.21	-0.45
VD	0.74	-0.15	0.46	-052	-0.07
PS	-0.48	-0.08	-0.07	-001	-0.02
DPZ	-0.24	-0.15	0.06	-0.22	-0.07

Table 2: PCA feature extraction.

PC	Feature value	Variance contribution rate (%)	Cumulative variance contribution rate (%)
PC1	3.753	36.958	36.958
PC2	2.241	29.505	66.463
PC3	1.273	25.458	91.921
PC4	0.673	4.802	96.723
PC5	0.528	3.277	100

way. Then, the greedy promotion algorithm was used to further update the dual variables to realize the online learning process of the S3VM model. This section focuses on the introduction of the OS^{3}VM algorithm based on the dual lifting process.

3.1. S^{3}VM Model. The objective optimization function of traditional SVM can be expressed as follows [36]:

$$J(m) = \frac{1}{2}\|m\|_2 + c\sum_{t=1}^{T}\sigma_t[1 - y_t\langle m, x_t\rangle], \tag{1}$$

where c represents the weight parameter, x_t represents the sample data, y_t represents the sample label, $\sigma_t = 1$ represents the marked sample, and $\sigma_t = 0$ represents the unmarked sample.

S^{3}VM extends the idea of maximizing the classification interval to semisupervised learning and comprehensively considers the role of marked samples and unmarked samples when maximizing the interval. The hat loss function is usually used in S^{3}VM to describe the loss caused by unmarked data:

$$L_t(m) = [1 - |\langle m, x_t\rangle|]. \tag{2}$$

Then, the objective function of S^{3}VM can be expressed as follows:

$$J(m) = \frac{1}{2}\|m\|_2 + \sum_{t=1}^{T}(c\sigma_t[1 - y_t\langle m, x_t\rangle]) + \gamma(1 - \sigma_t)[1 - |\langle m, x_t\rangle|], \tag{3}$$

where c and γ represent the weight parameters.

In order to adapt to online learning, this paper adopted a balanced penalty function to relax the constraint of the objective function on the decision boundary, and the objective function of S^{3}VM can be described as follows:

$$\begin{aligned} J(m) = {} & \frac{1}{2}\|m\|_2 + \sum_{t=1}^{T}(c\sigma_t[1 - y_t\langle m, x_t\rangle]) \\ & + \gamma(1 - \sigma_t)[1 - |\langle m, x_t\rangle|]) \\ & + \mu\sum_{t=1}^{T}\left|\frac{1}{u_t}\sum_{i=t-\tau+1}^{t}(1 - \sigma_i)\langle m, x_i\rangle - \frac{1}{l_t}\sum_{i=t-\tau+1}^{t}\sigma_i y_i\right|, \end{aligned} \tag{4}$$

where c, γ, and μ represent the weight parameters, $P_t \subseteq \{1, 2, \cdots, t\}$, u_t, and l_t are the number of unlabeled samples and labeled samples subscripted in P_t, respectively, and τ represents the size of P_t.

Equation (4) can be further simplified as follows:

$$J(m) = \frac{1}{2}\|m\|_2 + \sum_{t=1}^{T} w_t(m), \tag{5}$$

where $\sum_{t=1}^{T} w_t(m)$ represents the loss function of the sample (x_i, y_i, σ_i).

The sample loss function $w_t(m)$ can be decomposed into the sum of convex function $w_t^{\text{vex}}(m)$ and concave function $w_t^{\text{ave}}(m)$, and the objective function of S^{3}VM can also be decomposed into the sum of convex function $J^{\text{vex}}(m)$ and concave function $J^{\text{ave}}(m)$:

$$J(m) = \frac{1}{2}\|m\|_2 + \sum_{t=1}^{T} w_t(m) = J^{\text{vex}}(m) + J^{\text{ave}}(m) + \text{con}, \tag{6}$$

where con is a constant term, which will not affect the solution of minimizing $J(m)$, and this term can be ignored.

The optimal boundary vector can be defined as $m^* = \underset{m}{\operatorname{argmin}} J(m)$; by combining Equations (4) and (6), it can be obtained as follows:

$$(J^{\text{ave}})'(m^*) = \sum_{t=1}^{T} (w_t^{\text{ave}})'(m^*). \tag{7}$$

The predicted label based on the boundary vector m^* of the unlabeled sample x_t was defined as $\bar{y}_t = \text{sign}(\langle m, x_t \rangle)$. If there was a soft label $\tilde{y}_t$ approaching the predicted label $\bar{y}_t$, then the S^{3}VM model can be achieved by minimizing the following equation:

$$J^{\text{vex}}(m) - \gamma \sum_{t=1}^{T} \tilde{y}_t \langle m, x_t \rangle. \tag{8}$$

In this paper, the latest boundary vector obtained in the previous learning process was used as the soft label of the unmarked sample.

3.2. OS^{3}VM Algorithm Based on the Dual Lifting Process. The OS^{3}VM algorithm mainly includes two processes. The first is the prediction of the soft labels of unlabeled samples, and the second is the lifting process of the dual function [37]. The dual function corresponding to Equation (8) can be expressed as follows:

$$\begin{aligned} &D(\varphi_1, \varphi_2, \cdots, \varphi_T, \theta) \\ &= -\frac{1}{2}\left\{\sum_{t=1}^{T}\left[c\sigma_t\varphi_{t1}y_t x_t + \gamma(1-\sigma_t)(\varphi_{t2} - \varphi_{t3} + \tilde{y}_t)x_t - \mu\theta_t \frac{1}{u_t}\sum_{i\in P_t}(1-\sigma_i)x_i\right]\right\}^2 \\ &+ \sum_{t=1}^{T}\left[c\sigma_t\varphi_{t1} + \gamma(1-\sigma_t)(\varphi_{t2} + \varphi_{t3}) - \mu\theta_t \frac{1}{l_t}\sum_{i\in P_t}\sigma_i y_i\right], \end{aligned} \tag{9}$$

where $\varphi_t = [\varphi_{t1}, \varphi_{t2}, \varphi_{t3}] (t \in \{1, 2, \cdots, T\})$ represents the system vector corresponding to the loss function, and satisfies the constraint conditions $\varphi_{t1}, \varphi_{t2}, \varphi_{t3} \in [0, 1]$; $\theta = [\theta_1, \theta_2, \cdots, \theta_T]$ represents the coefficient vector corresponding to the penalty function and satisfies constraint conditions $\theta_1, \theta_2, \cdots, \theta_T \in [-1, 1]$.

Equation (9) indicates that the variables in the dual function corresponding to Equation (8) are actually a set of coefficient variables with constraints, and the function value of the dual function can be determined according to the value of the coefficient vector group $(\varphi_1, \varphi_2, \cdots, \varphi_T, \theta)$. In addition, since the coefficient variables in the dual function $D(\varphi_1, \varphi_2, \cdots, \varphi_T, \theta)$ are independent of each other, the function value of the whole dual function can be improved only by changing the value of some coefficient variables, so as to solve the OS^{3}VM model update problem.

$(\varphi_i)_t$ and $(\theta)_t$ represent the value coefficient vectors $\varphi_i (i \in \{1, 2, \cdots, t\})$ and θ in the learning period t. According to the characteristics of the data flow in the online semisupervised learning process, the coefficient vector group $(\varphi_1, \varphi_2, \cdots, \varphi_T, \theta)$ should meet the following four conditions in addition to its own constraints during the learning cycle update process [38]:

(1) For any $i \in \{1, 2, \cdots, T\}$, $(\varphi_{i1})_t = 0$

(2) For any $i \in \{t+1, t+2, \cdots, T\}$, $(\varphi_{i2})_t - (\varphi_{i3})_t + \tilde{y}_i = 0$

(3) For any $i \in \{t+1, t+2, \cdots, T\}$, $(\theta_i)_t = 0$

(4) The new dual vector group can improve the function value of the dual function, namely,

$$D\left((\varphi_1)_t, (\varphi_2)_t, \cdots, (\varphi_t)_t, (\theta)_t\right) \geq D\left((\varphi_1)_{t-1}, (\varphi_2)_{t-1}, \cdots, (\varphi_t)_{t-1}, (\theta)_{t-1}\right). \tag{10}$$

According to the above analysis, the boundary vector m_t of the learning period t in the OS^{3}VM algorithm can be expressed as follows:

$$\begin{aligned} m_t = \sum_{i=1}^{t}\Bigg[&c\sigma_i(\varphi_{i1})_t y_i x_i + \gamma(1-\sigma_i)(\varphi_{i2} - \varphi_{i3} + \tilde{y}_i)x_i \\ &- \mu(\theta_i)_i \frac{1}{u_i}\sum_{j\in P_t}(1-\sigma_j)x_j\Bigg]. \end{aligned} \tag{11}$$

Considering the computational complexity and punishment function design, the OS^{3}VM algorithm proposed in this paper only used the samples marked in $I_t = \{t - \tau + 1, t - \tau + 2, \cdots, t\}$ to achieve dual promotion process in the learning cycle t. Therefore, the set of dual variables that can be updated in the learning period was $\{\varphi_{t-\tau+1}, \varphi_{t-\tau+2}, \cdots, \theta_t\}$. At the same time, considering that the greedy promotion process would bring a greater degree of dual function promotion in each learning cycle, the paper proposed an OS^{3}VM algorithm based on greedy promotion. In the learning period t, the value of the dual function can be

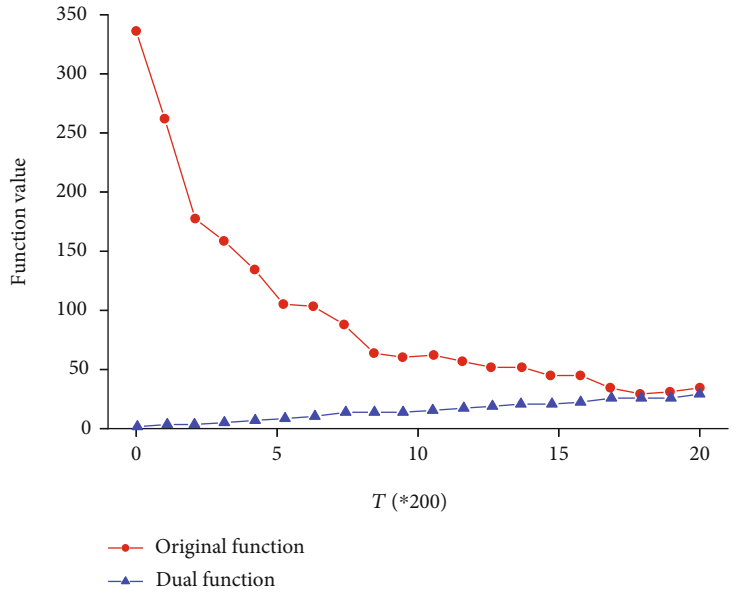


Figure 10: The change curve of the original function and the dual function.

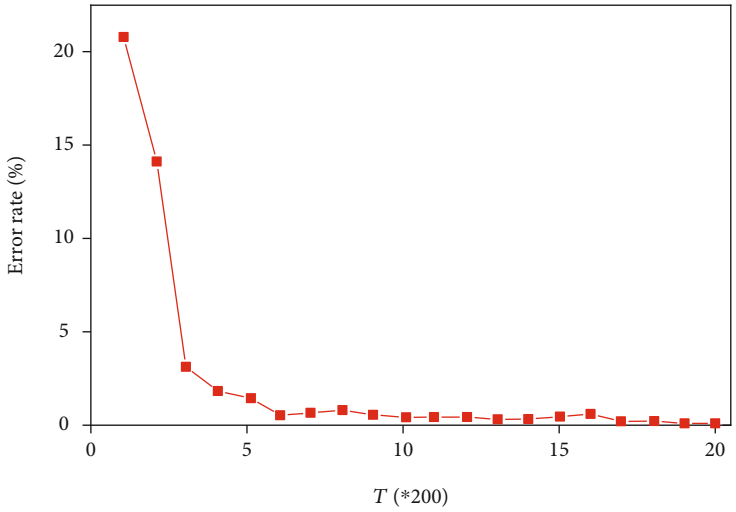


Figure 11: The error rate of boundary vectors in the whole data set.

improved by solving the following quadratic programming (QP) problem [24]:

$$\left((\varphi_1)_t, (\varphi_2)_t, \cdots, (\varphi_t)_t, (\theta)_t\right) = \underset{\varphi_2 \in [0,1]^3, \theta_i \in [-1,1]}{\operatorname{argmax}} D(\varphi_1, \varphi_2, \cdots, \varphi_t, \theta)$$
$$\text{s.t.} \forall i \notin P_t, \varphi_i = (\varphi_i)_{t-1}, \forall i \neq t, \theta_i = (\theta_i)_{t-1}. \tag{12}$$

Therefore, Equation (12) would have a maximum dual promotion in the dual variable set $\{\varphi_{t-\tau+1}, \varphi_{t-\tau+2}, \cdots, \theta_t\}$ after given a soft label, thus achieving the semisupervised support vector machine online learning process.

4. Simulation Results

In this article, we used MATLAB language for modelling; a total of 1800 groups of experimental data samples were obtained and randomly selected 10% to 70% of the samples (in 10% increments) without replacement as labeled samples l_t and then the remaining samples as unlabeled samples u_t. In addition, 30% of the unmarked samples u_t were randomly determined as the test sample. The SVM algorithm and S^3VM algorithm were employed to compare with the OS^3VM algorithm proposed in this study. The difference between a supervised model and a semisupervised model is that the SVM only uses the labeled samples l_t for training, while the S^3VM uses both marked samples l_t and unmarked samples

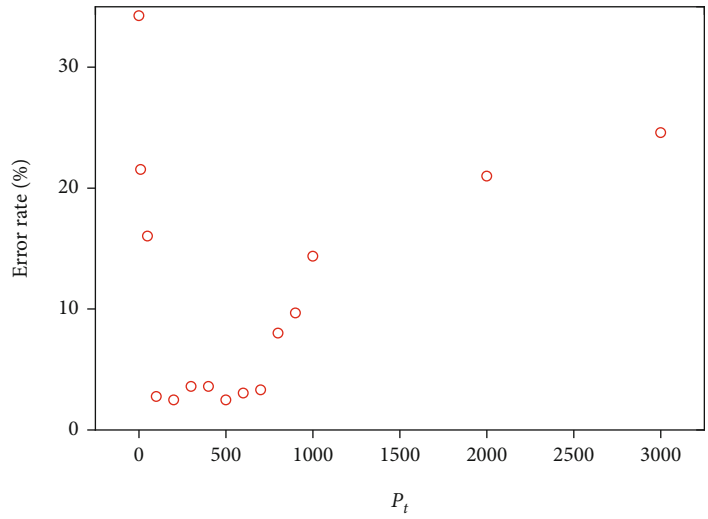


Figure 12: The impact of the size of P_t on the error rate of the algorithm.

u_t for training. Based on the S^{3}VM model, the OS^{3}VM can update the trained model in real time. In order to process the large data streams in practical applications, it is also necessary to sparse the boundary vectors in the process of the OS^{3}VM algorithm. Therefore, k-maximum dual coefficient (k-MC, $k = 400$) method was selected as the sparsification method in the experimental process. Since the OS^{3}VM algorithm only contains the inner product operation between sample points, the kernel function could also be introduced to find the linear classification surface. In this study, the standard RBF kernel function was determined to find the linear classification surface and its form can be expressed as follows:

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma_K^2}. \quad (13)$$

4.1. Dual Lifting Process. In order to further illustrate the effect of the dual lifting process, Figure 10 compares the change curves of the original function and the dual function in the OS^{3}VM algorithm. It can be seen from Figure 11 that the two curves are constantly approaching each other along with the algorithm process. The value of the dual function increases during the OS^{3}VM algorithm. In contrast, the value of the original function tends to decrease, and there will be some small fluctuations during the descending process. The results demonstrate the correctness of the proposed OS^{3}VM algorithm and the feasibility and effectiveness of the algorithm based on the dual promotion process. In addition, Figure 10 further reveals the error rate of boundary vectors $m(t)$ in the process of the OS^{3}VM algorithm in the whole data set. The simulation result indicates that the proposed algorithm is a process of improving the performance of the predictor. Since the algorithm only uses the information of local samples to update the predictor, the effect of the predictor will inevitably produce some small fluctuations during the learning process.

In the OS^{3}VM algorithm, P_t not only controls the number of local sample points used in the balanced penalty function to punish the unbalanced division but also determines the range of sample points used in the dual promotion process, and therefore also controls the computational time complexity of the algorithm process in each learning cycle. Obviously, the larger the size of P_t is, the higher the computational time complexity of the algorithm process in each learning cycle is. Figure 12 shows the impact of the size of P_t on the error rate of the algorithm. The results demonstrate that if the τ value is too small or too large, the classification effect of the algorithm will be worse. If the τ value is too large, the large number of sample points used in the dual lifting process will hardly express the sparse region of the current sample distribution, thus making the learning effect worse. In other words, too large a value of τ will make the algorithm unable to respond to changes in the data stream in time.

Based on the above analysis, the reasonable choice of P_t in the OS^{3}VM algorithm will be beneficial to improve the computational time complexity and the classification accuracy of the algorithm. Therefore, in the paper, the value of τ was determined as 200.

Table 3: Accuracy of pedestrian crossing intention recognition based on SVM, S3VM, and OS3VM.

Labeled sample proportion	SVM	S^{3}VM	OS^{3}VM
10	82.43	85.93	89.16
20	84.57	87.42	91.38
30	86.12	88.98	92.47
40	87.86	90.50	94.83
50	88.32	91.07	94.66
60	88.45	91.21	94.72
70	89.27	91.43	94.65

4.2. Comparison of Recognition Accuracy. The comparison results of pedestrian crossing intention recognition accuracy of SVM, S^{3}VM, and OS^{3}VM are shown in Table 3 and Figure 13. The results indicated that, in the case of the same

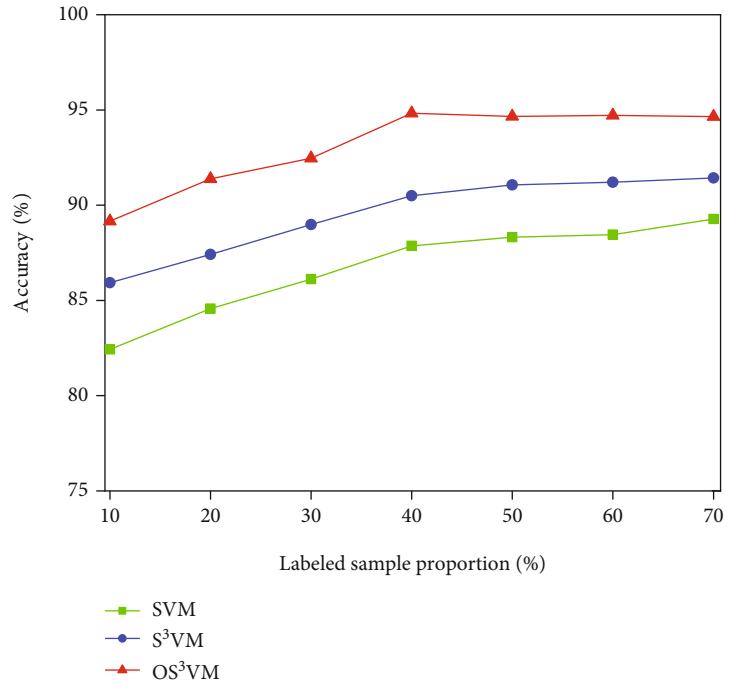


Figure 13: The comparison results of recognition accuracy of SVM, S3VM, and OS3VM.

proportion of labeled samples, the OS^{3}VM algorithm exhibits a better ability to recognize pedestrian crossing intentions than the SVM and S^{3}VM models. Under the condition of 10% marked sample ratio, the accuracy of pedestrian crossing intention recognition of OS^{3}VM is higher than of the SVM model with 60% marked sample ratio and the S^{3}VM with 30% marked sample ratio, and the accuracies are 89.16%, 88.45%, and 88.98%, respectively. It can be seen that the OS^{3}VM algorithm can improve the recognition ability of pedestrian crossing intention by using the unmarked samples. In addition, when the proportion of labeled samples is greater than 40%, the established model tends to converge and the accuracy rate converges to 94%. As the proportion of labeled samples increases, the training time of the model will greatly increase. Therefore, comprehensively considering the training time and recognition accuracy, the paper determines that the best-labeled sample ratio of the OS^{3}VM recognition model is 40%.

5. Conclusions

Autonomous vehicles need to understand pedestrian behavior in order to achieve better performance. Recognizing the pedestrian intention is one of the most critical capabilities for autonomous vehicles to ensure the safe operation of the urban environment. However, for autonomous vehicles, it is quite challenging to accurately identify pedestrians' crossing intentions, because they are affected by their emotions, traffic environment, road environment, and weather. At present, pedestrian crossing intention models have the problem that the models cannot be updated online in real time, which limits their applicability and generalization. To accurately identify pedestrians' crossing intentions, the model needs to be able to update the model online in real time according to the diversity of the samples. To achieve this goal, this paper proposes a OS^{3}VM. In order to verify the effectiveness of the model, this paper uses laser scanner to collect a large amount of pedestrian crossing data and vehicle movement data, and determines the input feature parameters of the model through statistical analysis and PCA feature extraction.

The semisupervised support vector machine is a type of semisupervised learning method based on low-density region segmentation. This type of method believes that the decision-making area should be located in some areas with low data density. Since the solution of S^{3}VM algorithm is a nonconvex problem and difficult to handle, there is relatively little research work on online semisupervised support vector machine. This study proposes an OS^{3}VM model based on dual promotion process to identify the pedestrian crossing intentions. Firstly, the hat loss function is employed to describe and define the basic learning problem of the S^{3}VM model. Then according to the inspiration in the process of concave and convex, the nonconvex problem would be transformed to a convex problem. Therefore, an OS^{3}VM model based on dual promotion process is established with using the greedy algorithm to achieve the improvement of the dual function. Finally, the online update of the classifier is completed. In order to verify the validity of the proposed algorithm, the SVM and S^{3}VM models are established, respectively, and the accuracy of pedestrian crossing intention recognition of different models is compared under the different labeled sample proportions. The results demonstrate that the proposed OS^{3}VM model can reduce the number of labeled samples for training the classifier and improve the recognition accuracy.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Key Research and Development Program of Shaanxi under Grant 2020GY-173 and in part by the Fundamental Research Funds for the Central Universities, CHD 300102220220.

References

[1] Traffic Administration Bureau of the Ministry of Public Security of the People's Republic of China, *Road Traffic Safety Development Report*, Traffic Administration Bureau of the Ministry of Public Security of the People's Republic of China, Wuxi, China, 2019.

[2] W. Chen, X. Zhuang, Z. Cui, and G. Ma, "Drivers' recognition of pedestrian road-crossing intentions: performance and process," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 64, pp. 552–564, 2019.

[3] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[4] A. Rasouli and J. K. Tsotsos, "Autonomous vehicles that interact with pedestrians: a survey of theory and practice," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 900–918, 2020.

[5] B. Yang and R. Ni, "Vision-based recognition of pedestrian crossing intention in an urban environment," in *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 992–995, Suzhou, China, 2019.

[6] S. Kalantarov, R. Riemer, and T. Oron-Gilad, "Pedestrians' road crossing decisions and body parts' movements," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 53, pp. 155–171, 2018.

[7] A. T. Schulz and R. Stiefelhagen, "A controlled interactive multiple model filter for combined pedestrian intention recognition and path prediction," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 173–178, Las Palmas, 2015.

[8] D. Varytimidis, F. Alonso-Fernandez, B. Duran, and C. Englund, "Action and intention recognition of pedestrians in urban traffic," in *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 676–682, Las Palmas de Gran Canaria, Spain, 2018.

[9] B. Völz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto, "A data-driven approach for pedestrian intention estimation," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2607–2612, Rio de Janeiro, Brazil, 2016.

[10] K.-H. Park and S.-W. Lee, "Movement intention decoding based on deep learning for multiuser myoelectric interfaces," in *2016 4th International Winter Conference on Brain-Computer Interface (BCI)*, pp. 1-2, Yongpyong, South Korea, Feb. 2016.

[11] H. Zhang, Y. Liu, C. Wang, R. Fu, Q. Sun, and Z. Li, "Research on a pedestrian crossing intention recognition model based on natural observation data," *Sensors*, vol. 20, no. 6, p. 1776, 2020.

[12] F. Schneemann and P. Heinemann, "Context-based detection of pedestrian crossing intention for autonomous driving in urban environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2243–2248, Daejeon, South Korea, 2016.

[13] J. Zhao, Y. Li, H. Xu, and H. Liu, "Probabilistic prediction of pedestrian crossing intention using roadside LiDAR data," *IEEE Access*, vol. 7, pp. 93781–93790, 2019.

[14] Z. Fang, D. Vázquez, and A. López, "On-board detection of pedestrian intentions," *Sensors*, vol. 17, no. 10, 2017.

[15] J. Škovierová, A. Vobecký, M. Uller, R. Škoviera, and V. Hlaváč, "Motion prediction influence on the pedestrian intention estimation near a zebra crossing," in *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems*, pp. 341–348, Madeira, Portugal, 16–18 March 2018.

[16] R. Quintero Mínguez, I. Parra Alonso, D. Fernández-Llorca, and M. Á. Sotelo, "Pedestrian path, pose, and intention prediction through Gaussian process dynamical models and pedestrian activity recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, 2018.

[17] R. Quintero, I. Parra, J. Lorenzo, D. Fernández-Llorca, and M. A. Sotelo, "Pedestrian intention recognition by means of a hidden Markov model and body language," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–7, Yokohama, 2017.

[18] W. Shaohua, X. Xiaolong, W. Tian, and G. Zonghua, "An intelligent video analysis method for abnormal event detection in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[19] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, New York, 2006.

[20] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.

[21] C. Wang, Q. Sun, Y. Guo, R. Fu, and W. Yuan, "Improving the user acceptability of advanced driver assistance systems based on different driving styles: a case study of lane change warning systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4196–4208, 2020.

[22] S. Ding, S. Qu, Y. Xi, and S. Wan, "A long video caption generation algorithm for big video data retrieval," *Future Generation Computer Systems*, vol. 93, pp. 583–595, 2019.

[23] A. B. Goldberg, M. Li, and X. Zhu, "Online manifold regularization: a new learning setting and empirical study," in *Machine Learning and Knowledge Discovery in Databases*, pp. 393–407, Springer, Berlin, Heidelberg, 2008.

[24] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof, "On-line semi-supervised multiple-instance boosting," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010.

[25] J. Zhao, J. O. Malenje, Y. Tang, and Y. Han, "Gap acceptance probability model for pedestrians at unsignalized mid-block

crosswalks based on logistic regression," *Accident; Analysis and Prevention*, vol. 129, pp. 76–83, 2019.

[26] B. R. Kadali and P. Vedagiri, "Proactive pedestrian safety evaluation at unprotected mid-block crosswalk locations under mixed traffic conditions," *Safety Science*, vol. 89, pp. 94–105, 2016.

[27] D. Herrero-Fernández, P. Macía-Guerrero, L. Silvano-Chaparro, L. Merino, and E. C. Jenchura, "Risky behavior in young adult pedestrians: personality determinants, correlates with risk perception, and gender differences," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 36, pp. 14–24, 2016.

[28] K. V. R. Ravishankar and P. M. Nair, "Pedestrian risk analysis at uncontrolled mid-block and unsignalized intersections," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 5, no. 2, pp. 137–147, 2018.

[29] X. Zhuang and C. Wu, "Modeling pedestrian crossing paths at unmarked roadways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1438–1448, 2013.

[30] Y. Hashimoto, Y. Gu, L. Hsu, and S. Kamijo, "Probability estimation for pedestrian crossing intention at signalized crosswalks," in *2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 114–119, Yokohama, 2015.

[31] 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.

[32] H.-P. Deutsch, M. W. Beinker, H.-P. Deutsch, and M. W. Beinker, "Principal component analysis," in *Derivatives and Internal Models: Modern Risk Management*, pp. 793–804, Palgrave Macmillan, Cham, 2019.

[33] C. M. Martinez, M. Heucke, F. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, 2018.

[34] X. J. Zhu, *Semi-supervised learning literature survey*, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[35] L. Wang, H. Zhen, X. Fang, S. Wan, W. Ding, and Y. Guo, "A unified two-parallel-branch deep neural network for joint gland contour and segmentation learning," *Future Generation Computer Systems*, vol. 100, pp. 316–324, 2019.

[36] O. Chapella and A. Zien, "Semi-supervised classification by low density separation," *Machine Learning*, 2004.

[37] S. Shalev-Shwartz and Y. Singer, "A primal-dual perspective of online learning algorithms," *Machine Learning*, vol. 69, no. 2-3, pp. 115–142, 2007.

[38] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6691348, 13 pages
https://doi.org/10.1155/2021/6691348

Research Article

Nonintrusive Load Management Based on Distributed Edge and Secure Key Agreement

Jing Zhang[1], **Qi Liu**[1], **Lu Chen**[2], **Ye Tian**[3] and **Jun Wang**[1]

[1]*School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China*
[2]*Department of Information Security, Naval University of Engineering, Wuhan 430033, China*
[3]*China Information Communication Technologies Group Corporation (CICT), Wuhan 430205, China*

Correspondence should be addressed to Lu Chen; ieucl@163.com

Received 13 November 2020; Revised 30 December 2020; Accepted 9 January 2021; Published 29 January 2021

Academic Editor: Sotirios K. Goudos

With the advancement of national policies and the rise of Internet of things (IoT) technology, smart meters, smart home appliances, and other energy monitoring systems continue to appear, but due to the fixed application scenarios, it is difficult to apply to different equipment monitoring. At the same time, the limited computing resources of sensing devices make it difficult to guarantee the security in the transmission process. In order to help users better understand the energy consumption of different devices in different scenarios, we designed a nonintrusive load management based on distributed edge and secure key agreement, which uses narrowband Internet of things (NB-IoT) for transmission and uses edge devices to forward node data to provide real-time power monitoring for users. At the same time, we measured the changes of server power under different behaviors to prepare for further analysis of the relationship between server operating state and energy consumption.

1. Introduction

In the new era, the Internet has fundamentally changed social life, and people's demand for the Internet is also increasing. A new generation of network communication reform is emerging. The IoT can be regarded as the extension of the Internet, that is, the Internet extends its tentacles to the field of embedded computer systems and their supporting sensors, connecting all objects to the network through information sensing devices such as QR codes, radio frequency identification, and sensors, so as to transmit information, so as to form a worldwide interconnected mode to realize automatic identification, precise positioning, real-time tracking, and timely management [1]. At present, IoT technology has been widely used in agriculture, retail, logistics, storage, medical, energy, and other fields. Figure 1 shows the ecological map of the IoT [2]. Due to the impact of global warming, as well as the trend of diversified social energy use, global energy consumption continues to increase; energy conservation and emission reduction have become an important topic in the world [3]. NB-IoT technology as a high security, high-quality, low-power, and low-cost IoT technology has been applied to energy management field more and more [4]. The existing energy management pays more attention to providing users with equipment status monitoring, household energy consumption statistical analysis, differential electricity price information, and other services through the monitoring of electric energy information [5]. Energy management not only provides users with visual information statistics but also allows users to get more information and sense of participation, so as to encourage users to actively save energy and improve household energy utilization rate [6]. In addition, the statistics of relevant electric energy information can also provide the basis for the smart allocation and pricing of the State Grid and improve the security and reliability of power grid operation [7].

The energy management and control system based on IoT mainly uses wireless transmission mode. Different application scenarios involve different communication modes, including Bluetooth, ZigBee, WiFi, GSM/3G/4G cell, and HTTP. Due to the use scenarios, different sensing means must be used. However, due to the limited memory and computing power of a large number of sensing devices in the basic layer, it is difficult to achieve security defence, and the

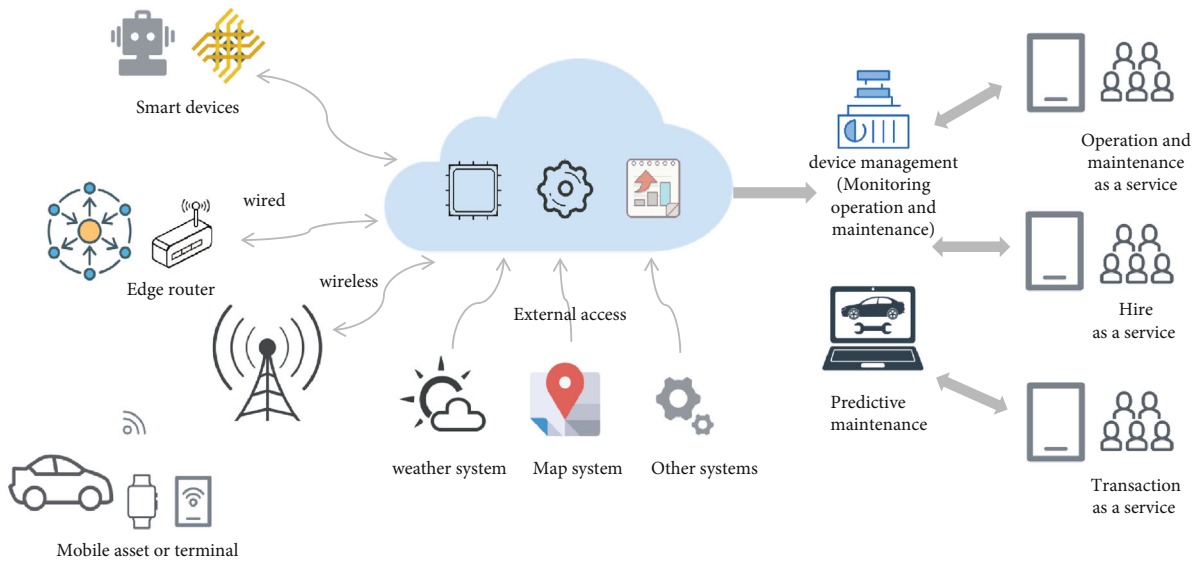


Figure 1: The ecosystem of IoT. The ecosystem includes equipment, platform, application, and business. After the platform is connected to the equipment, it can be used to build and manage IoT solutions for different industries.

data exposed in the public environment is very easy to steal and is easy to be interfered with [8]. Therefore, how to maintain the system data security and reliability is a problem we need to consider [9]. The IoT can be divided into sensing layer composed of sensors and other sensing devices, network layer responsible for data transmission, and application layer for visual display. Due to the variety of sensors in the sensing layer and the complexity of data format, it is difficult to completely unify, so the security control of network layer transmission is more complex [10]. Due to the limited resources of the sensing layer devices, only a small amount of computation can be performed. The deployment of classical encryption and authentication algorithms will not only consume equipment energy but also occupy resources and reduce the computing efficiency of devices. The existing security algorithms are mainly aimed at remote user authentication to prevent illegal users from gaining access to privacy data and controlling intelligent devices [11]. We designed a nonintrusive load management based on distributed edge and secure key agreement to provide real-time power monitoring for users. At the same time, we designed a lightweight encryption authentication algorithm between cloud edge devices to maintain the cloud server's receiving of sensor data and ensure the operation security of the energy management system.

After the second part introduces the related technology of the energy management system, we introduce the design of the secure energy management system in the third part. Then, the system performance is discussed in the fourth part. In the fifth part, we summarize our nonintrusive load management based on distributed edge and secure key agreement.

2. Related Work

The characteristics of IoT connecting a variety of sensors and sensing devices bring indispensable convenience to users' life and industrial production. In order to realize intelligent irrigation and save the cost of water resources, Rao et al. designed the adaptive control algorithm of a home irrigation system based on IoT. The water demand of plants was calculated by using the data of temperature and humidity sensor, and the water pump was controlled to irrigate plants in time to promote plant growth [12]. In the logistics and transportation industry, in order to ensure food safety, Gialelis and others designed a food traceability platform using low-cost IoT nodes to monitor the logistics chain from the "loading point" and continuously monitor the product storage environment [13]. In the health care industry, the IoT technology also has a variety of applications. Onasanya and Elshakankiri proposed a cancer medical system based on the IoT, which monitors the status of patients and environmental data through implanted and nonimplanted sensors, so as to provide timely and detailed information feedback for follow-up treatment, so as to help patients get better treatment and nursing [14]. In power, the IoT is mainly used in large-scale power grid and household small switch control. At present, the existing power Internet of things products mainly include smart meters and household appliances with wireless control module [15], while the equipment transmission mode mainly includes Bluetooth, WiFi, ZigBee, and NB-IoT. Adiono et al. proposed a Bluetooth-based smart home Android Software to help users control the power switch, lighting, curtain, door lock, and other devices in the home and monitor the temperature and humidity status in real time [16]. Madhu and Vyjayanthi designed a smart home controller using WiFi networking and controlled the corresponding equipment through the running software on the smart phone [17]. Jhang et al. designed a smart home control device based on ZigBee, and the sensor realized remote monitoring of door opening and closing and water leakage [18]. Due to its short communication distance and less connection number, Bluetooth technology is more suitable for the transmission of short

distance and less devices, such as wearable devices and small audio [19]. Because most of the intelligent products on the market, such as mobile phones, laptops, and TV boxes, have the function of accessing the Internet through WiFi, many products in the field of energy monitoring use WiFi as wireless communication mode. However, its configuration is complex, its security is low, and its power consumption is high, so it needs to be charged frequently. It is not applicable to sensor networks that need to transmit a small amount of data and contain a large number of nodes [20]. However, NB-IoT has the characteristics of low power consumption, large number of connections, wide transmission range, and low cost, so it has a great application prospect [21].

With the extensive use of NB-IoT technology, the user experience and product security need to be improved. For example, once the data of intelligent meter reading is stolen, it can be inferred whether the household is at home according to the data, which leads to the risk of property loss. At present, the existing IoT solutions not only face the problem of bandwidth delay but also face challenges in resource constraints and security. Because the extra security increases the cost of most manufacturers rather than the profit of equipment sales, many manufacturers give up providing security patches and firmware update services. Based on this, there are many high-risk vulnerabilities in the current Internet of things devices, especially in the default password, plaintext transmission key, and so on [22]. As shown in Figure 2, the IoT system is mainly composed of three layers, and each layer has different security vulnerabilities due to different technologies used [23]. Generally, the front-end devices of the sensing layer are limited by resources and cannot carry out complex computing tasks, so it is difficult to protect data security. Therefore, Batalla and Gonciarz designed a security algorithm deployed on edge devices [24]. Unde and Deepthi analyzed the rate distortion of compressive sensing (CS) using structural random matrix (SRM), injected artificial noise into the quantization CS measurement to resist CPA, and proposed a lightweight cryptography system based on compressed sensing for IoT, which reduced the computational burden and effectively reduced the complexity of CS encoder [25]. The energy management and control system based on the IoT may face active attacks and passive attacks, among which active attacks mainly include tampering and forgery, while passive attacks mainly include eavesdropping and deception, which are easy to cause a large amount of perceived information and user privacy information leakage [26]. Rehman and Gruhn developed a security algorithm to establish a sicher firewall between the software system and the smart home network as a filter to protect the system from virus attacks and unauthorized access [27]. Lyu et al. designed an antitracking mutual authentication scheme deployed on the ifttt server to achieve the anonymity of data transmission and ensure system security [28]. Naoui et al. proposed a user authentication scheme with additional security functions, which can resist multiple attacks such as internal attacks and simulation attacks and improve the security of user authentication [29].

We design a load management system with cloud edge architecture, which places the computing requirements of the sensing layer on the edge devices, which can effectively save the computing cost of the sensing layer and reduce the system delay. Meanwhile, we adopt a lightweight access control algorithm to ensure the communication security between the edge device and the cloud server and reduce the risk of perceived data and user information leakage. In addition, we use the load management system to test the power consumption of the server under different actions, so as to further analyze the relationship among the server power consumption with its running state.

3. The Proposed Secured Nonintrusive Load Management System

The nonintrusive secure load management system is designed to monitor, store, and process the power consumption of the detected equipment in real time. Compared with intrusive monitoring, the nonintrusive system is easy to install, so the system adopts nonintrusive design. Considering the limited storage and computing resources of monitoring nodes, the system adopts the cloud edge architecture design and uses edge devices to package and forward, which can reduce the computing cost of monitoring nodes. Meanwhile, compared with protecting the data being processed, it is usually much easier to protect the data transmitted through the network, and the system reliability is higher. At the same time, we use a lightweight access control algorithm to ensure the reliability of process transmission without affecting the transmission efficiency.

The system is mainly composed of monitoring nodes, edge devices, cloud servers, and clients. The overall design structure is shown in Figure 3. The system consists of three parts, including the perceptual layer composed of monitoring nodes, the transport layer consists of edge devices and cloud servers, and the application layer. Wireless sensor network adopts tree topology structure, with cloud server as the core, supplemented by multiple edge devices and monitoring nodes. Take the power management of the server room as an example. When the system works, the monitoring node regularly reads the power consumption data of the server and then sends messages to the edge device. The power data is packaged and encrypted by the edge device and transmitted to the cloud server through the wireless communication network. Users can view and manage network nodes through visual web pages and application programs in the application layer and access the power consumption data of corresponding servers to prepare for subsequent data analysis.

3.1. Nonintrusive Metering Module. The nonintrusive metering module is devoted to monitor the power consumption value of the monitored equipment at the current moment and forward the messages to the edge devices. The overall structure is shown in Figure 4. The core of the module is the STC15W404S chip, with power monitoring module, external memory, NB-IoT transmission module, and some peripheral circuits.

The main control unit adopts STC15W404AS single-chip microcomputer with an enhanced 8051 core produced by Hongjing Technology company to control data storage and

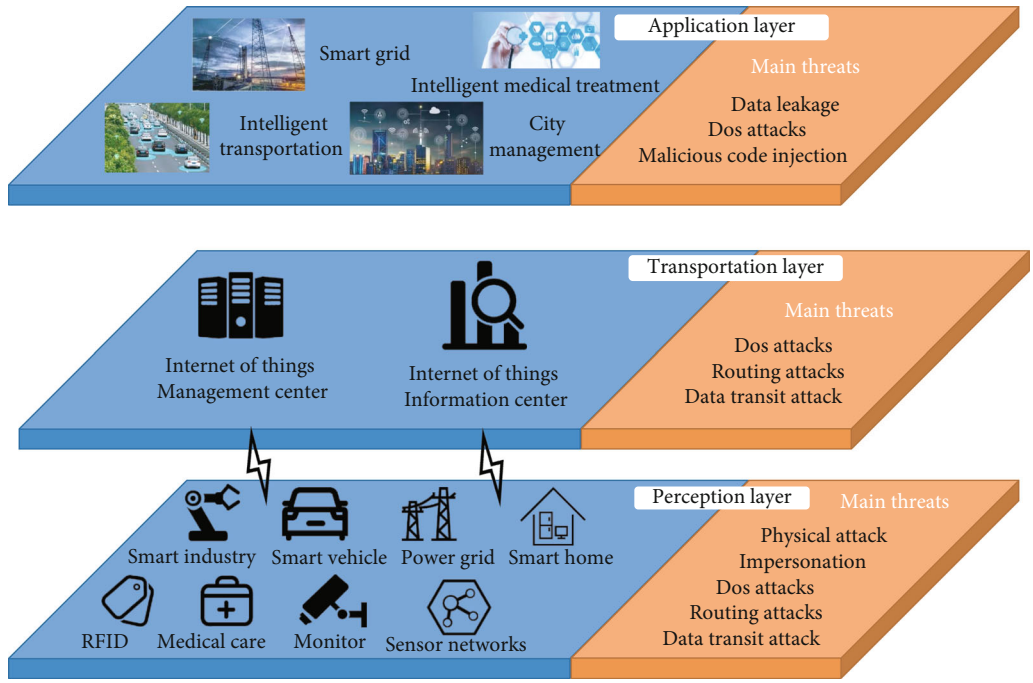


Figure 2: Threats in the IoT system model. Threats of perception layer: physical attack, impersonation, dos attacks, routing attacks, and data transit attack. Threats in the transport layer: dos attacks, routing attacks, and data transit attack. Threats of application layer: data leakage, dos attacks, and malicious code injection.

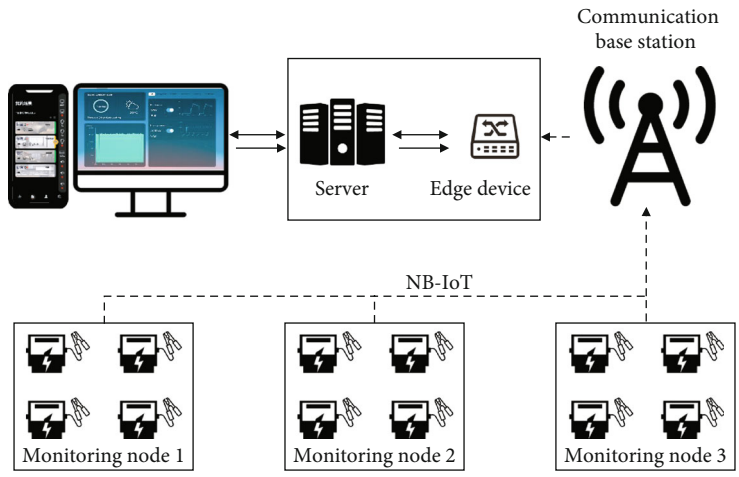
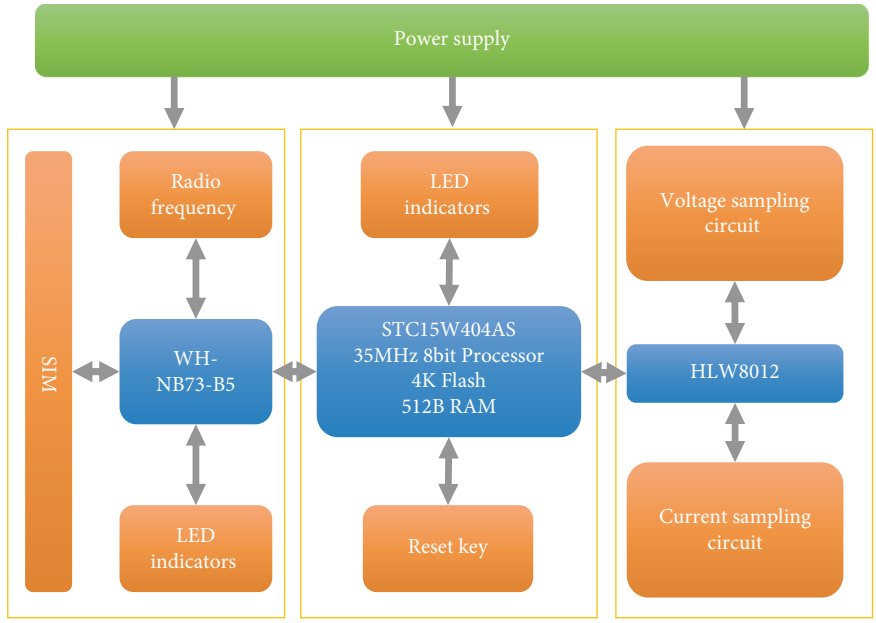
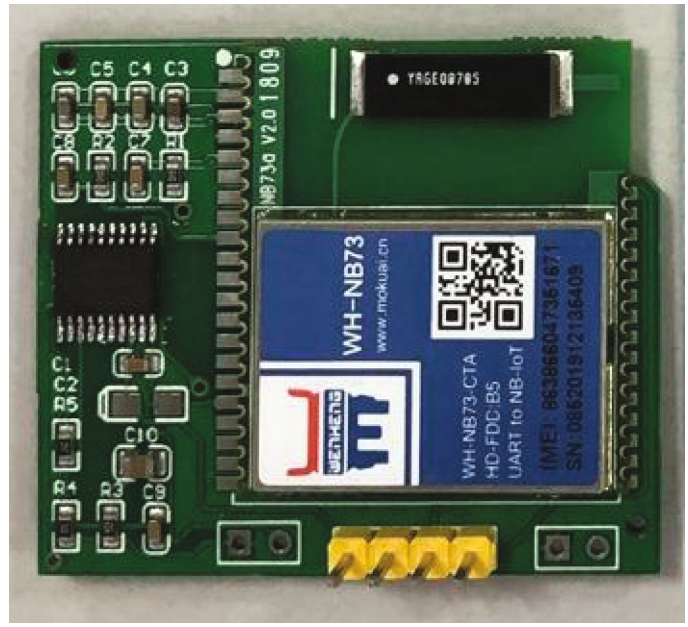


Figure 3: System architecture. The system includes monitoring node, edge device, cloud server, and client. The data is collected and transmitted to the edge device by the monitoring node and then packaged and sent to the cloud server for display at the client.

transmission. The chip not only has low power consumption and low price but also has an ultra-high-speed CPU core with the highest frequency of 35 MHz. The chip is driven by an internal crystal oscillator without an external crystal oscillator and reset. It contains a high-speed asynchronous serial interface (UART) and has rich pin functions. It can also be connected with 74HC595 to expand the general I/O port. The current and voltage sampling methods of the electric energy metering module are divided into transformer sampling and resistance sampling. The monitoring node adopts the isolation method to monitor the power consumption value. The 2 mA/2 mA current type voltage transformer cooperates with the resistance to convert the voltage signal into the current signal meeting the input conditions in the voltage acquisition circuit. In the current sampling circuit, the current transformer with the transformation ratio of 1000 : 1 combined with the sampling resistance converts the measured current into a low-voltage signal, which is also input into the power monitoring chip through the filter circuit. NB-IoT has the benefit of low power consumption, high security, large number of devices allowed to be connected, and communication distance of more than 10 km, which



(a)



(b)

Figure 4: Design of monitoring node circuit: (a) PCB and block diagram of monitoring node; (b) prototype of control module and sensor.

can be well applied to most IoT application scenarios. Therefore, the WH-NB73 module produced by Shanghai Wenheng Technology Co., Ltd. is adopted as the communication module. The power metering module and MCU are powered by 5 V DC voltage, and the NB-IoT communication module WH-NB73-B5 is powered by 3.3 V. The monitoring node adopts the AC-DC power module WA3-220S05A3 to convert 220 V AC to 5 V DC. At the same time, the node uses the AMS 117-3.3 DC voltage regulator chip of AMS company to realize the DC voltage stabilizing function from 5 V to 3.3 V.

3.2. Cloud-Edge-Node Architecture System. The most important work of the cloud server is to accept the electricity messages from users and store them into the persistent database after aggregation. The data analysis and mining process will analyze the data from users in the cloud, push the analysis results to the client application, and respond to the client data request. The data is measured and transmitted to the edge devices by the sensing layer monitoring nodes. The edge devices are packaged and encrypted and sent to the cloud server. Before the message is stored, the cloud server processes and analyses the data, and then, users can access the data of each node in real time.

3.2.1. Node and Edge Communication. TCP communication mode is used in the system sensing layer and network layer

communication. After the monitoring node is powered on, MCU and energy metering module are initialized. At the same time, the network is searched and added. The power consumption data and node address of the monitored equipment are read regularly and packaged into data packets and sent to edge devices. After receiving the message, the edge device judges whether it is the target node according to the address and Sn in the message. If so, it will process the message; otherwise, the message will be ignored. Considering reducing the energy consumption of the node, if there is no event to be handled, the monitoring node will automatically enter into the sleep state. The sleep includes the NB-IoT communication module sleep and the power monitoring module sleep. When the system finishes regularly and needs to read and send monitoring data, it wakes up the two sleep modules and only wakes up the communication module when processing other events. The main process of monitoring node program is shown in Algorithm 1.

```
Begin
Initialize the resource
Search for network, Read SSID and Password
While Successfully joined the network
    Start timing
    If have an event
        End timing
        Read power
        Send data
    Else if system sleep
Endwhile
End
```

Algorithm 1: Monitoring node algorithm.

The monitoring node communicates with the edge device in TCP mode, and the data packet is transmitted in JSON format, as shown in Table 1. For the data package uploaded by the monitoring node, the edge device first judges the packet type, then analyzes whether the node address matches with Sn, processes the monitoring data, and caches it. In a certain time interval, the received message is packaged and encrypted and transmitted to the cloud server to protect the security in the transmission process.

Table 1: Monitoring node upload data protocol.

Name	Data type	Mean
Type	char	The data type of this data
ID	int	Equipment number
SN	char	Serial number of the device
Power	float	Instantaneous electric power
Time	int	Current time

3.2.2. Cloud and Edge Communication. The program of the edge device is written in Java, and the NIO nonblocking communication is realized by using the Java Mina communication framework, which improves the communication performance of the edge device under the condition of high concurrency. After receiving the monitoring node, the edge device packs and encrypts the node and forwards it to the cloud server. After adding the corresponding node with the configuration function on the WeChat applet and application program, the user can access the corresponding node data, such as node number, real-time energy consumption, and historical statistics. The energy management system provides users with visual information statistics, which makes users more convenient and efficient to obtain equipment energy consumption and equipment operation status, timely handle abnormal operation equipment, save energy, and ensure safe operation of equipment. The logical architecture of device cloud and edge communication is shown in Figure 5.

The communication between cloud edges mainly includes three operations: connection, transmission, and disconnection. The specific process is as follows:

The connection function process is divided into four steps: (1) the edge device initiates the connection, and the API gateway establishes the connection and records the ID; (2) the API gateway notifies the edge device that the connection is established; (3) triggers the connection function to run and transfers the connection ID; (4) records the connection ID to the database and changes the device online status.

The transfer function process is divided into three steps: (1) the edge device initiates the message; (2) the API gateway transmits the message and triggers the transfer function; (3) the information in the message is extracted and stored in the database.

The process of disconnection function is divided into three parts: (1) the edge device initiates the disconnection request; (2) the API gateway triggers the disconnection function and transfers the connection ID; (3) queries the corresponding ID in the database and changes the online status.

3.2.3. Secure Communication Protocol. In the energy management system of IoT, data delay and reliability are the important judgment basis of the system. Pei et al. compared and analyzed the memory consumption and avalanche effect of six high-performance lightweight block ciphers and found that speck has the best comprehensive performance [30]. Therefore, based on the nonintrusion load management system, we use a speck algorithm to ensure the safety of message transmission. Here, we briefly introduce the speck algorithm.

The speck series algorithm is a kind of lightweight block cipher algorithm proposed by the national security agency of the United States. The algorithm adopts the deformed Feistel structure, and the round function is the ARX component. It is composed of mixed operations of modular integer addition, cyclic shift, and XOR operation. The main nonlinear operation is modular integer addition. Speck algorithm is more flexible than other algorithms. It supports 32, 24, 64, 96, and 128 bit blocks. The round function of the speck series algorithm is shown in Table 2.

Speck 2n/mn is used to represent speck algorithm with a packet length of 2n bit and key length of mn bit, where $n \in \{16, 24, 32, 48, 64\}$, $m \in \{2, 3, 4\}$. Remember the algorithm master key $K = (L_{m-2}, L_{m-3}, \cdots, L_0, K_1)$, where $K_0, L_0 \in$

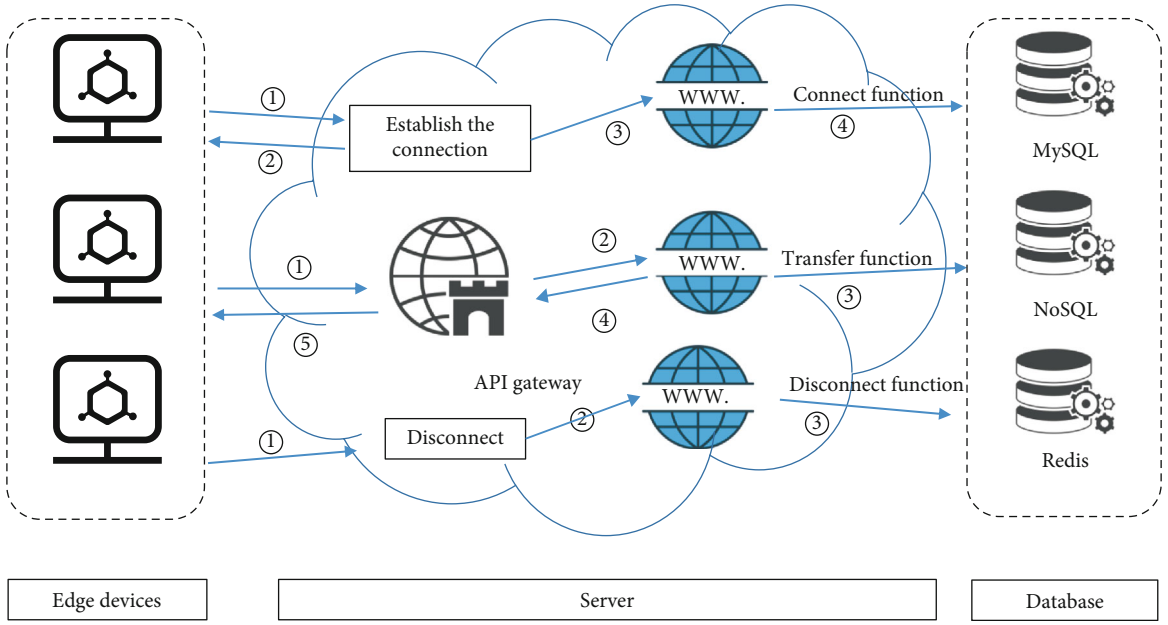


Figure 5: Logical process of cloud edge. Data is transferred and processed between edge devices, servers, and databases. The communication between cloud edges mainly includes three operations: connection, transmission, and disconnection.

Table 2: Speck series algorithm version.

Block size	Key size	Speck rounds	n	m
32	64	22	16	4
48	72	22	24	3
48	96	23	24	4
64	96	26	32	3
64	128	27	32	4
96	96	28	48	2
96	144	29	48	3
128	128	32	64	2
128	192	33	64	3
128	256	34	64	4

$\{0, 1\}^n$, and m is the number of key blocks of each algorithm. The key expansion algorithm is

$$\begin{cases} l_{i+m-1} = (K_i + l_i >>> \alpha) \oplus i, \\ K_{i+1} = (K_i <<< \beta) \oplus l_{i+m-1}. \end{cases} \quad (1)$$

Output N subkey $K_0, K_1, \cdots, K_{N-1}$.

4. Experiments and Performance Evaluation

4.1. Experiments. After the system design is completed, the actual deployment is tested. The system server is deployed on the Tencent cloud platform, and the API gateway is used for data interaction. Tencent cloud server with CPU of 2.3 GHz, 1 GB memory, and Ubuntu 16.04.1 LTS operating system is used to forward messages instead of edge devices. The monitoring nodes are deployed on the power lines of 5 different electrical equipment, connected to the power supply, waiting for the equipment networking, and 10 simulators are turned on at the same time. When the system works, bind the device name, device ID, and device location on the applet configuration interface, as shown in Figure 6. The main interface displays the current device name, current power, and total load power consumption in a list mode, and the historical data of the corresponding device can be viewed in the scene interface; also, we can configure it in the metainterface.

4.2. Feasibility Analysis. We use the monitoring equipment to measure the energy consumption changes of the server under different behaviors. The results show that the energy consumption waveform of the server is similar under the consent action, and the waveform diagram is shown in Figure 7. We divide the behavior into three categories: U-disk operation, network communication, and office software operation. Among them, U-disk operation includes U-disk plug-in and file transmission, network communication includes communication software start-up and shutdown, network transmission file, and office software operation includes office start-up and shutdown. When measuring the changes of server energy consumption, we record the performance changes of server CPU, memory, and network, which provide the basis for us to further finding the relationship among server power consumption with performance changes.

In (i) U-disk insertion operation and (ii) U-disk pull-out operation in Figure 7(a), the plug-in edge event characteristics are very obvious, and the peak energy consumption increases between 30 and 45 W, and the peak value of pulling out the U-disk is generally lower than that of inserting the U-disk. In the operation of transferring files to a USB flash disk, as shown in Figure 7(b), power consumption increases during transmission, but the fluctuation law is not obvious. As shown in Figure 7(c), QQ software operation (i) on and (ii)

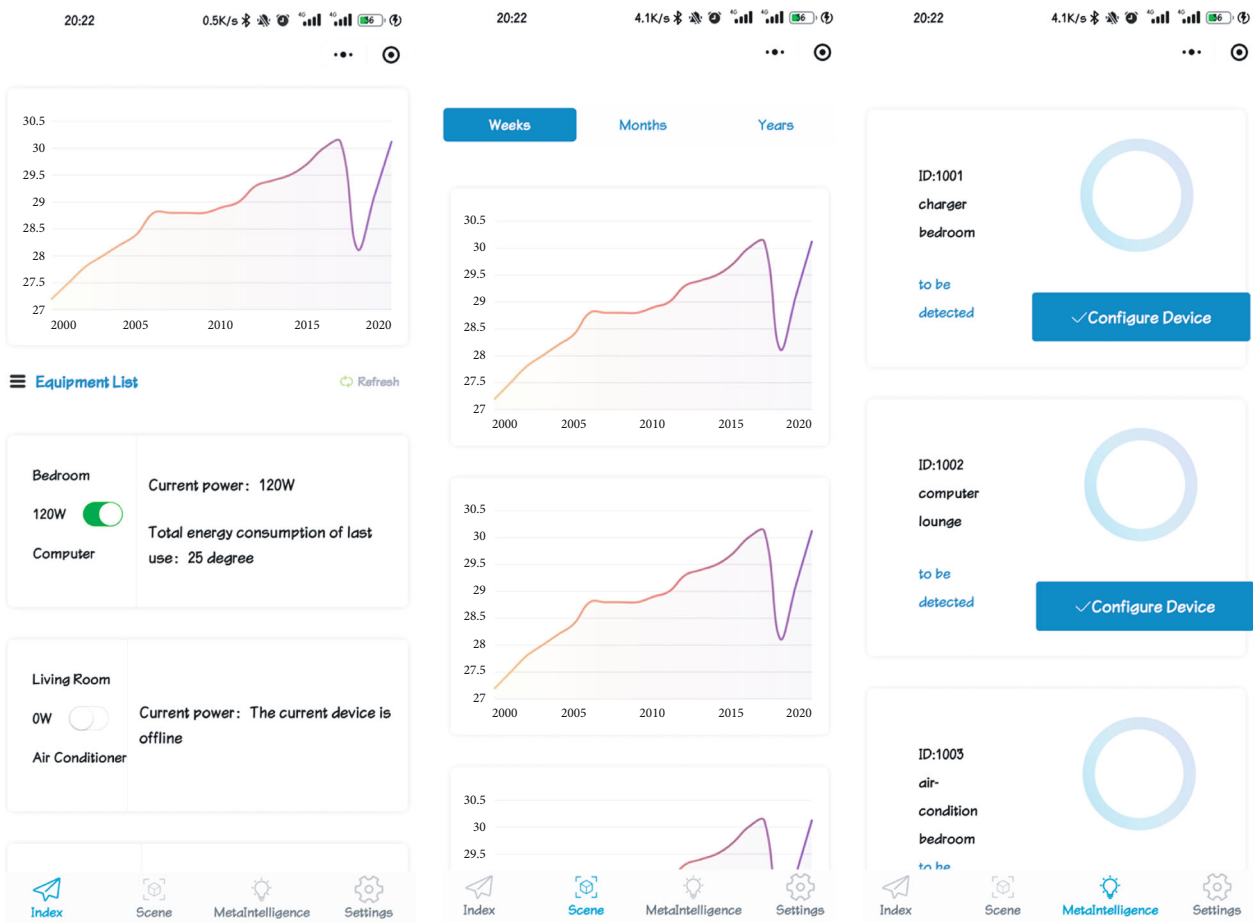
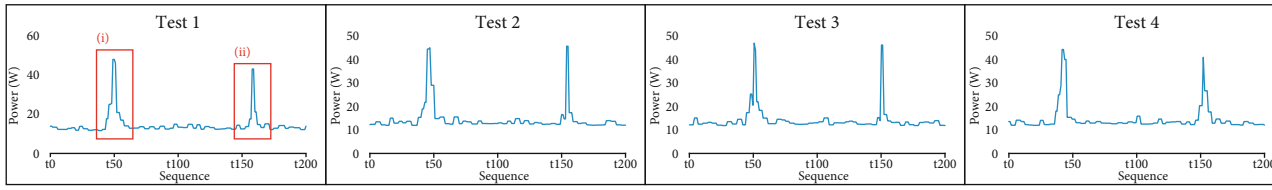


Figure 6: Main interface of the applet: (a) main interface; (b) scene interface; (c) metaintelligence interface.

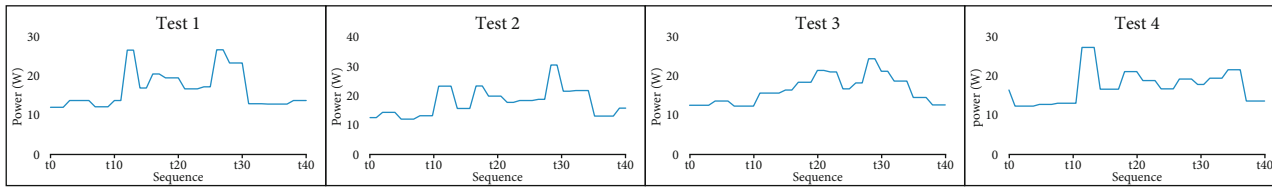
off edge event characteristics are obvious, and the waveform changes are regular. The opening event generally lasts 50~60 ms and tends to be flat after three main peak fluctuations, and the first peak fluctuates the most and then shows a decreasing trend. The closure events generally last for 25~40 ms and tend to be flat after 2~3 main peak fluctuations, and the first peak fluctuation is the smallest and then presents an increasing trend. In the process of QQ transmission, as shown in Figure 7(d), there is no obvious energy consumption fluctuation. As shown in Figure 7(e), WeChat software operation (i) on and (ii) off edge event characteristics are obvious, and the waveform changes are regular. The opening event generally lasts 15~20 ms and tends to be flat after 1~2 main peak fluctuations, and the former fluctuation is small. The closing event usually lasts 15~20 ms, and the peak fluctuation is less than the maximum fluctuation of the opening operation. In order to use WeChat software to transfer files, as shown in Figure 7(f), the energy consumption increases significantly and fluctuates regularly. At the beginning, there will be a small fluctuation, and then, the energy consumption will last for a period of 40~60 W. As shown in Figure 7(g), for PDF file (i) open and (ii) close operation, edge event characteristics are obvious and regular. In conclusion, it is feasible to infer the current event based on the change of server energy consumption. Then, we further measure the relationship between the server energy consumption change and the performance changes of CPU, memory, and network under specific events.

As shown in Figure 8, there is a correlation between the change of server energy consumption and the change of CPU, memory, network, and other performance under specific events. The correlation between CPU utilization and energy consumption is the most significant, which is the main factor affecting the change of energy consumption. In the first column, the mutation of the energy consumption curve is very similar to the mutation of CPU utilization, and the change of memory is similar. In the second column, the curve of CPU is closely related to the curve of energy consumption. The third column of QQ open and close operation and the fourth column of WeChat transfer files have the same rules. At the same time, we found that the change of the network operating with chat software is also full of rules, and different software has its own characteristics. Therefore, it is feasible to infer the current server performance change or even the current running software according to the server energy consumption change. In the next work, we will further explore the relationship between them and energy consumption.

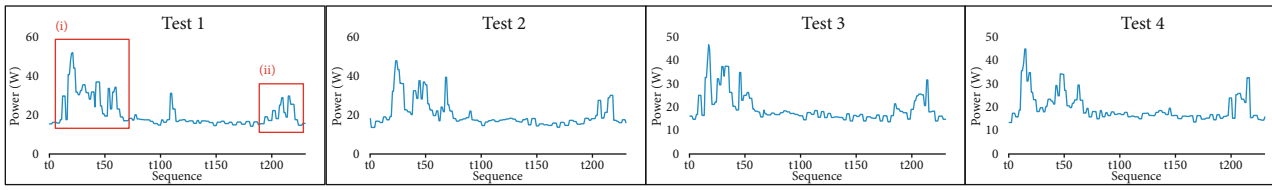
4.3. Performance Analysis. In this paper, a nonintrusive load management based on distributed edge and secure key



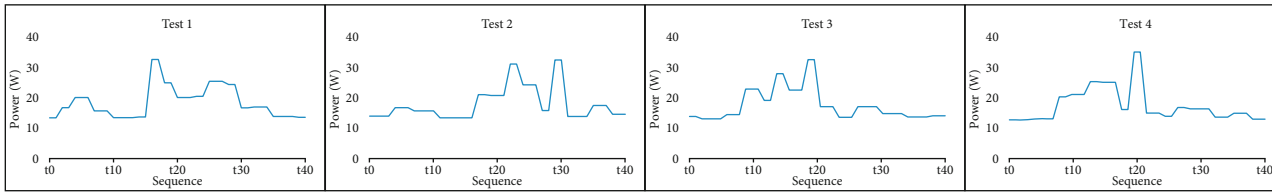
(a)



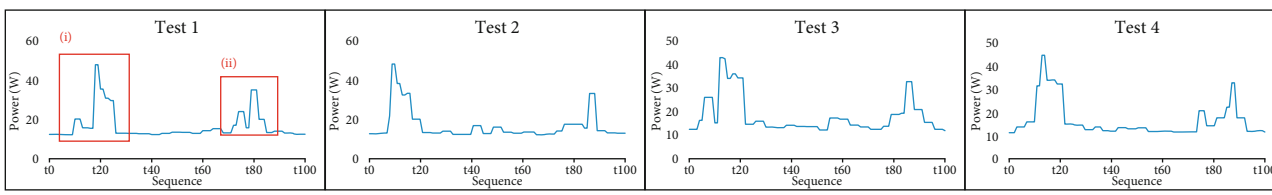
(b)



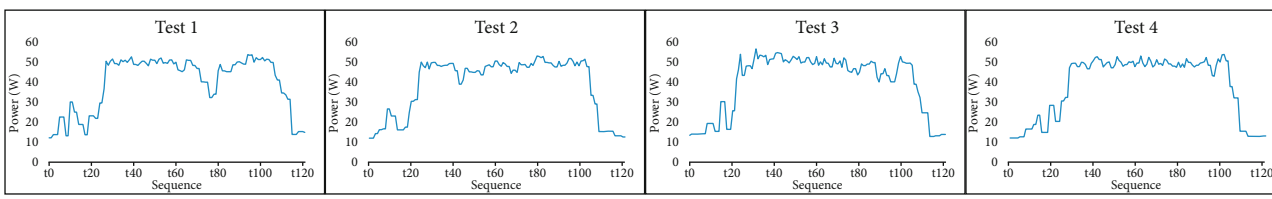
(c)



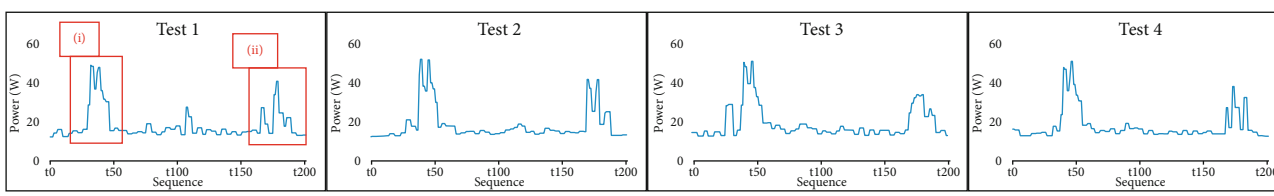
(d)



(e)



(f)



(g)

Figure 7: Energy consumption waveform: (a) U-disk plug and pull; (b) disk transfer files; (c) open and close QQ software; (d) use QQ to transfer files; (e) open and close WeChat software; (f) use WeChat to transfer files; (g) open and close PDF files.

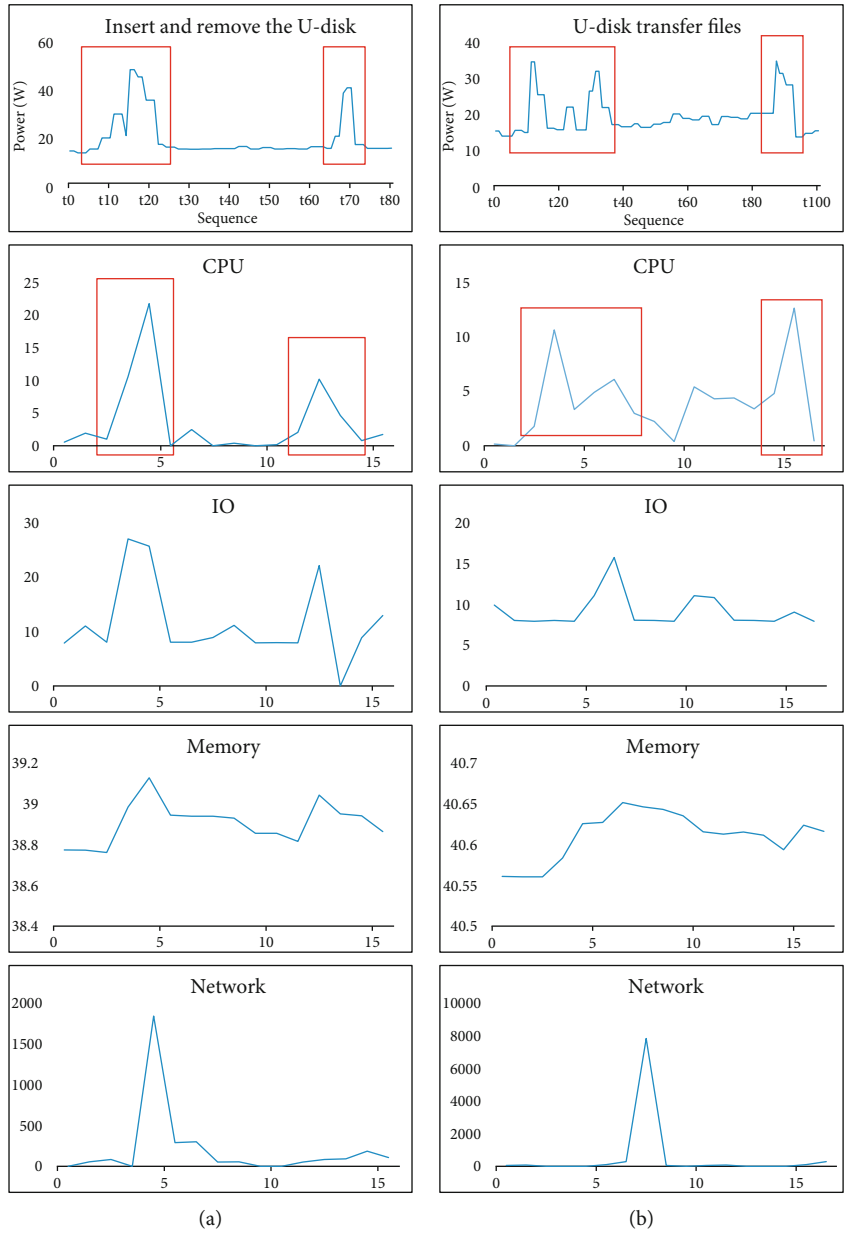


Figure 8: Continued.

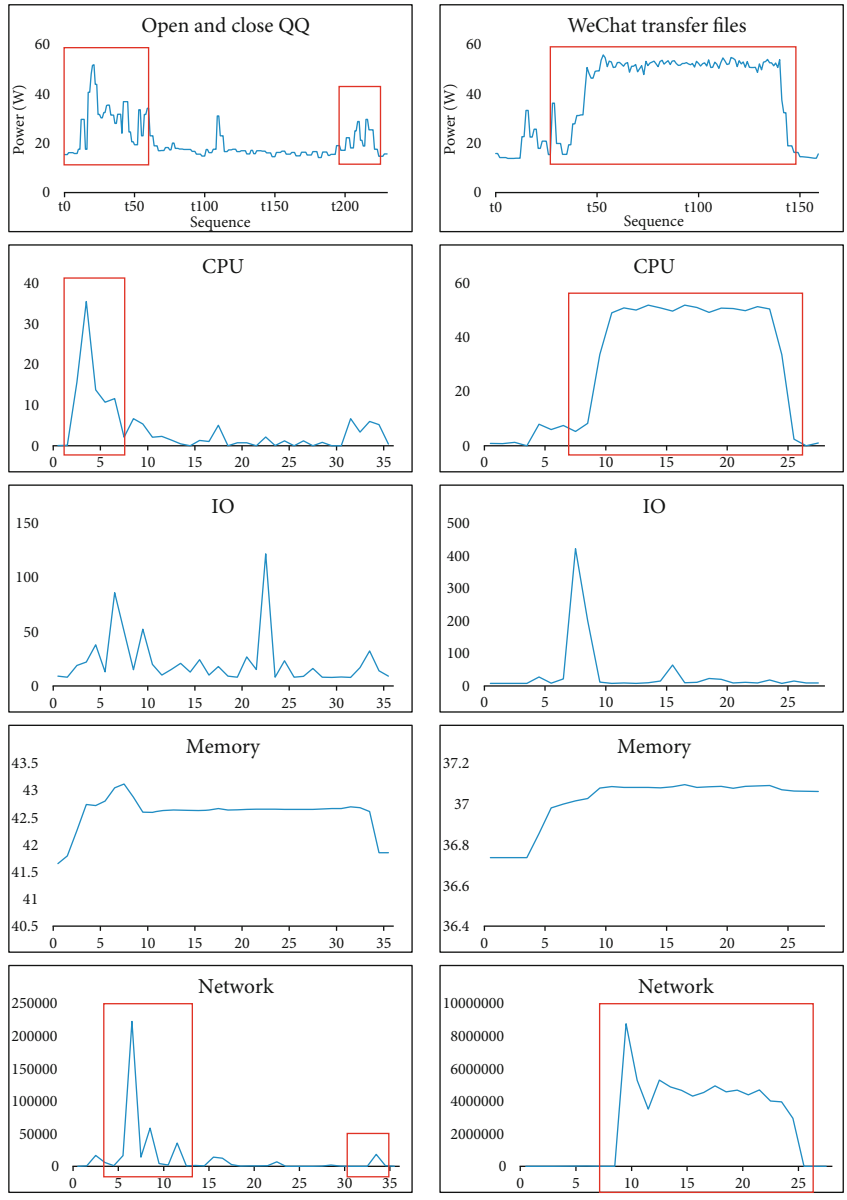


Figure 8: Energy consumption and the relationship among CPU, IO, memory, and network. The first column is insert and remove the U-disk, the second column is U-disk transfer file, the third column is open and close QQ, and the fourth column is WeChat transmission file.

agreement is designed. Compared with other energy consumption monitoring systems, the system uses NB-IoT as communication module, which has longer communication distance, better signal strength, and higher security. The non-intrusive calliper monitoring design can well meet the application requirements of various occasions and is easy to install. Meanwhile, the architecture design of cloud edge reduces the computing resources of monitoring nodes, which is convenient for deploying higher security encryption algorithm to ensure system security. Speck lightweight encryption algorithm used in the system takes less computing resources, can resist various types of attacks, and has a good security margin.

In our experiment, we run the monitor node simulator on a personal computer (HP with an inter (R) core (TM) i7-7700hq @ 2.80 GHz 2.81 GHz processor, 16 GB main memory, and window 10 operating system) and an app on a personal mobile device (Huawei nova5 pro with quad-core 2.6 g processor, 8 GB memory, and Android 10 operating system) as a user. We have done this 4000 times to get the average run time.

Table 3: Data transfer time (MS).

Operations	T_1	T_2	T_3	T_4	T_5	T_{Avg}
User times	29.32	30.15	34.31	32.80	34.27	34.97

We test the time delay between the edge device and the cloud platform when the system is running. As shown in Table 3, we calculate the transmission delay and the

total average delay of five groups of devices. The minimum delay is 10 ms, the maximum delay is 193 ms, and the delay is concentrated between 25 and 35 ms to meet the transmission requirements, which is affected by the network changes.

5. Conclusions

Power data acquisition technology based on NB-IoT technology will be the main technical direction of 5 g technology applied in smart grid in the future, based on the characteristics of low power consumption and wide transmission range of NB-IoT; this paper designs a nonintrusive load management based on distributed edge and secure key agreement, which uses edge devices to encrypt and forward node data and accesses control algorithm to ensure system data security. In addition, this paper measured the server power change under different behaviors, and the results show that the waveform of server power change is similar under fixed behavior. Next, we plan to further analyze the relationship between energy consumption and server performance change by measuring the server CPU, memory, GPU, network, and energy consumption change data at the same time, so as to infer the abnormal state of the server by using the energy consumption change to provide managers with more detailed early warning.

Data Availability

Our processed data involve two parts. One is from 3rd party, i.e., REDD, which can be downloaded via this link: http://redd.csail.mit.edu/. The other data were generated in our own lab for testing and evaluating purposes by using smart sockets.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Jing Zhang and Qi Liu contributed equally to this work.

Acknowledgments

This work has received funding from the National Natural Science Foundation of China (Nos. 41911530242 and 41975142), 5150 Spring Specialists (05492018012 and 05762018039), Major Program of the National Social Science Fund of China (Grant No. 17ZDA092), 333 High-Level Talent Cultivation Project of Jiangsu Province (BRA2018332), Royal Society of Edinburgh, UK, and China Natural Science Foundation Council (RSE Reference: 62967_Liu_2018_2) under their Joint International Projects funding scheme, National Natural Science Foundation of China (Grant No. 41875184), Innovation Team of "Six Talent Peaks" in Jiangsu Province (Grant No. TD-XYDXX-004), and Basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20191398 and BK20180794).

References

[1] K. Chopra, K. Gupta, and A. Lambora, "Future internet: the internet of things-a literature review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 135–139, Faridabad, 2019.

[2] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta, and A. Bhatnagar, "CRAIoT: concept, review and application(s) of IoT," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1–4, Ghaziabad, India, 2019.

[3] S. S. Hosseini, K. Agbossou, S. Kelouwani, and A. Cardenas, "Non-intrusive load monitoring through home energy management systems: a comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1266–1274, 2017.

[4] Z. Qin, F. Y. Li, G. Y. Li, J. A. Mccann, and Q. Ni, "Low-power wide-area networks for sustainable IoT," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 140–145, 2019.

[5] O. Elma and U. S. Selamoğullar, "A survey of a residential load profile for demand side management systems," in *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, pp. 85–89, Oshawa, ON, Canada, 2017.

[6] L. Liu, Y. Liu, L. Wang, A. Zomaya, and S. Hu, "Economical and balanced energy usage in the smart home infrastructure: a tutorial and new results," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 4, pp. 556–570, 2015.

[7] G. Gaur, N. Mehta, R. Khanna, and S. Kaur, "Demand side management in a smart grid environment," in *2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC)*, pp. 227–231, Singapore, Singapore, 2017.

[8] E. R. Naru, H. Saini, and M. Sharma, "A recent review on lightweight cryptography in IoT," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 887–890, Palladam, India, 2017.

[9] Y. Chen, X. Wang, Y. Yang, and H. Li, "Location-aware Wi-Fi authentication scheme using smart contract," *Sensors*, vol. 20, no. 4, pp. 1062–1083, 2020.

[10] L. Li, F. Zhu, H. Sun, Y. Hu, Y. Yang, and D. Jin, "Multi-source information fusion and deep-learning-based characteristics measurement for exploring the effects of peer engagement on stock price synchronicity," *Information Fusion*, vol. 69, pp. 1–21, 2021.

[11] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 391–406, 2020.

[12] R. Y. Rao, J. J. Koola, N. D. Mehta, and A. M. Haque, "Design and implementation of adaptive control algorithm for IoT based domestic irrigation system," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, Kanpur, India, 2019.

[13] J. Gialelis, G. Theodorou, and C. Paparizos, "A low-cost internet of things (IoT) node to support traceability: logistics use case," in *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, pp. 72–77, Valencia, 2019.

[14] A. Onasanya and M. Elshakankiri, "Smart integrated IoT healthcare system for cancer care," *Wireless Networks*, vol. 2019, pp. 1–16, 2019.

[15] H. Ikezawa and M. Imafuku, "Convenience survey of IoT house equipment for a smart life," in *2020 IEEE 2nd Global*

Conference on Life Sciences and Technologies (LifeTech), pp. 290–294, Kyoto, Japan, 2020.

[16] T. Adiono, S. F. Anindya, S. Fuada, K. Afifah, and I. G. Purwanda, "Efficient android software development using MIT app inventor 2 for Bluetooth-based smart home," *Wireless Personal Communications*, vol. 105, no. 1, pp. 233–256, 2019.

[17] G. M. Madhu and C. Vyjayanthi, "Implementation of cost effective smart home controller with Android application using node MCU and internet of things (IOT)," in *2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE)*, pp. 1–5, Shillong, India, 2018.

[18] W. H. Jhang, L. Chen, W. Chang, C. Yang, and C. Yu, "Design of a low-cost level-triggered Zigbee network multi-application sensor in smart homes," in *2017 6th International Symposium on Next Generation Electronics (ISNE*, pp. 1–3, Keelung, Taiwan, 2017.

[19] H. Joh, I. Yang, and I. Ryoo, "The internet of everything based on energy efficient P2P transmission technology with Bluetooth low energy," *Peer-to-Peer Networking and Applications*, vol. 9, no. 3, pp. 520–528, 2016.

[20] A. A. Zaidan, B. B. Zaidan, M. Y. Qahtan et al., "A survey on communication components for iot-based technologies in smart homes," *Telecommunication Systems*, vol. 69, no. 1, pp. 1–25, 2018.

[21] L. Wan, Z. Zhang, and J. Wang, "Demonstrability of Narrowband Internet of Things technology in advanced metering infrastructure," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 2–12, 2019.

[22] L. Touati, H. Hellaoui, and Y. Challal, "Threshold yoking/-grouping proofs based on CP-ABE for IoT applications," in *2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 568–575, Tianjin, 2016.

[23] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: present and future challenges," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2483–2495, 2018.

[24] J. M. Batalla and F. Gonciarz, "Deployment of smart home management system at the edge: mechanisms and protocols," *Neural Computing and Applications*, vol. 31, no. 5, pp. 1301–1315, 2019.

[25] A. S. Unde and P. P. Deepthi, "Design and analysis of compressive sensing-based lightweight encryption scheme for multimedia IoT," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 167–171, 2020.

[26] U. Saxena, J. S. Sodhi, and Y. Singh, "Analysis of security attacks in a smart home networks," in *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, pp. 431–436, Noida, 2017.

[27] S. Rehman and V. Gruhn, "An approach to secure smart homes in cyber-physical systems/internet-of-things," in *2018 Fifth International Conference on Software Defined Systems (SDS)*, pp. 126–129, Barcelona, Spain, 2018.

[28] Q. Lyu, N. Zheng, H. Liu, C. Gao, S. Chen, and J. Liu, "Remotely access "my" smart home in private: an anti-tracking authentication and key agreement scheme," *IEEE Access*, vol. 7, pp. 41835–41851, 2019.

[29] S. Naoui, M. E. Elhdhili, and L. A. Saidane, "Lightweight and secure password based smart home authentication protocol: LSP-SHAP," *Journal of Network and Systems Management*, vol. 27, no. 4, pp. 1020–1042, 2019.

[30] C. Pei, Y. Xiao, W. Liang, and X. Han, "Trade-off of security and performance of lightweight block ciphers in industrial wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 134 pages, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6695696, 18 pages
https://doi.org/10.1155/2021/6695696

Research Article

Multistrategy Repeated Game-Based Mobile Crowdsourcing Incentive Mechanism for Mobile Edge Computing in Internet of Things

Chuanxiu Chi,[1,2] **Yingjie Wang**,[1,2] **Yingshu Li**,[1,3] **and Xiangrong Tong**[1,2]

[1]*School of Computer and Control Engineering, Yantai University, Yantai 264005, China*
[2]*Yantai Key Labortory of High-end Ocean Engineering Equipment and Intelligent Technology, Yantai University, Yantai 264005, China*
[3]*Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA*

Correspondence should be addressed to Yingjie Wang; wangyingjie@ytu.edu.cn

Received 11 November 2020; Revised 10 December 2020; Accepted 4 January 2021; Published 27 January 2021

Academic Editor: Shaohua Wan

With the advent of the Internet of Things (IoT) era, various application requirements have put forward higher requirements for data transmission bandwidth and real-time data processing. Mobile edge computing (MEC) can greatly alleviate the pressure on network bandwidth and improve the response speed by effectively using the device resources of mobile edge. Research on mobile crowdsourcing in edge computing has become a hot spot. Hence, we studied resource utilization issues between edge mobile devices, namely, crowdsourcing scenarios in mobile edge computing. We aimed to design an incentive mechanism to ensure the long-term participation of users and high quality of tasks. This paper designs a long-term incentive mechanism based on game theory. The long-term incentive mechanism is to encourage participants to provide long-term and continuous quality data for mobile crowdsourcing systems. The multistrategy repeated game-based incentive mechanism (MSRG incentive mechanism) is proposed to guide participants to provide long-term participation and high-quality data. The proposed mechanism regards the interaction between the worker and the requester as a repeated game and obtains a long-term incentive based on the historical information and discount factor. In addition, the evolutionary game theory and the Wright-Fisher model in biology are used to analyze the evolution of participants' strategies. The optimal discount factor is found within the range of discount factors based on repeated games. Finally, simulation experiments verify the existing crowdsourcing dilemma and the effectiveness of the incentive mechanism. The results show that the proposed MSRG incentive mechanism has a long-term incentive effect for participants in mobile crowdsourcing systems.

1. Introduction

The increasing data demand in the 5G era is a huge challenge for IoT devices with limited computing power and resources. All data are transmitted through the network to the traditional cloud platform for centralized processing. This method is not conducive to data security and privacy and has poor real-time performance and high bandwidth pressure and energy consumption [1, 2]. The emergence of MEC can effectively reduce the risk of privacy leakage and system delays and relieve the pressure on network bandwidth and data center energy consumption [3, 4]. Due to the limited resources and computing power of a single device in MEC, a large amount of sense data is needed in practical applications [5, 6]. Mobile crowdsourcing is considered a promising method for obtaining massive amounts of data [7].

Mobile crowdsourcing provides a new model for solving problems by gathering the wisdom of groups. Numerous crowdsourcing websites show the collaborative nature of crowdsourcing, such as Yahoo! Answers [8] and Wikipedia [9, 10], which can be viewed as systems where small tasks are performed in exchange for rewards. At present, crowdsourcing has been applied in many fields, such as environmental quality inspection issues, noise level detection

projects, and commercial map services [11–14]. Workers use various devices to sense the environment in MEC, and edge nodes replace traditional cloud platforms to process these transaction data [15, 16]. The success of a crowdsourcing system largely depends on whether users are actively involved and their willingness to make efforts for sensing tasks [17, 18]. Due to insufficient participants in perception tasks and data quality issues, crowdsourcing systems need to adopt appropriate incentive mechanisms to actively guide participants to participate in tasks and provide high-quality services [19, 20]. Since the number of workers cannot maintain long-term participation, it is important to encourage workers to maintain higher enthusiasm under the condition that the number of workers does not decrease. Besides, the issue of completion quality is also an important research issue of the incentive mechanism. Since the platform and the requester cannot directly observe the worker's status, there is no guarantee that the task will be completed with high quality [21–23]. Completing a task requires resources such as battery power, computing resources, and data traffic [24]. To obtain greater benefits, workers may be lazy to reduce their cost consumption, which will seriously affect the completion quality of the task.

In this paper, the incentive mechanism is added in the crowdsourcing process, and a reasonable pricing plan is formulated based on the contribution of workers to complete the task. To compensate for the user's direct sensing cost when performing a particular sensing task, we call it short-term sensing incentive, but long-term participation of workers requires long-term incentives. Therefore, this paper researches the long-term incentive mechanism to motivate workers to continuously improve their efforts and ensure that workers continuously provide high-quality data.

In order to solve the problem, this paper models the interaction between workers and requesters as repeated games and calculates the specific discount factor value that could maintain the equilibrium. In the repeated game, according to the worker's data contribution, the degree of effort is divided into five intervals, i.e., the worker's strategy is the five different degrees of effort. In this paper, an evolutionary game is introduced to simulate the evolution of players' strategic choices, and the evolutionary stability strategy (ESS) of both players is analyzed [25]. The simulation results show that the requester and the worker will eventually adopt the strategy of high effort and employment to maintain the long-term optimal benefits. In summary, the main contributions of this article are as follows:

(1) This paper uses a game to simulate the interaction between requester and workers, shows the conflict of interest between the two parties, and proves that there is a crowdsourcing dilemma in a dynamic game

(2) Dynamic strategy selection is modeled as an evolutionary game. The evolutionary game theory and the Wright-Fisher model are combined to analyze the evolution trend, and the Wright-Fisher model is used to calculate the adaptability of different populations to calculate the adaptability of users with different strategies

(3) This paper designs an incentive mechanism based on the multistrategy repeated game model to solve the dilemma. Using the discount factor of the repeated game and historical data, the current behavior of the two sides of the game is proposed, and an effective algorithm is proposed to obtain the Nash equilibrium and find the optimal payoff

(4) Both theoretical analysis and simulation experiments show that the proposed incentive mechanism could more effectively motivate workers to continue to provide high-quality data to improve the performance of the platform. The experimental results also show that under the proposed incentive mechanism, the steady state of the evolution of the two parties' strategies is no longer a dilemma state

The rest of the paper is organized as follows. Section 2 reviews the related works and presents the motivation for our work. Section 3 gives an overview of the crowdsourcing system and three definitions of the system. Section 4 describes the crowdsourcing dilemma and uses an evolutionary game to analyze the changes in the strategies of both players. Section 5 proposes a multistrategy model based on a repeated game and describes the algorithm. Section 6 presents our simulation results and analysis of the results. Conclusions and future work are discussed at the end.

2. Related Work

From the earlier discussion, we can discover that the workers will be unwilling to contribute information unless they receive enough compensation for their cost of resources. So the research on incentive mechanisms is necessary. In this section, we investigate and study the related works of incentive mechanism, repeated game theory, and evolutionary game theory for strategy analysis.

2.1. Incentive Mechanisms. An incentive mechanism is commonly used in crowdsourcing applications as a key part of the crowdsourcing system. Incentives are achieved by solving the problems of quality, payment control, and energy consumption when maximizing the utilities. There are many different incentive methods in the perception of swarm intelligence, which can be divided into monetary incentives and nonmonetary incentives [26–30]. Incentive mechanisms based on monetary rely on monetary or matching rewards in the form of micropayment to motivate workers to provide high-quality services [31]. The monetary incentive mechanism is mainly based on the game theory and auction mechanism [32, 33]. A reverse auction is different from the traditional auctions because the reverse auction includes one buyer and multiple sellers. During the reverse auction [25, 34–38], the requester selects a subset of workers with a lower cost under a certain budget consideration, and the sense data of workers is purchased at their bid prices. Considering that specific tasks are limited by budget and require workers with one or more skills, Zhang et al. [25] proposed a reverse auction-based incentive mechanism, assigning tasks to competent workers and rewarding workers for completing tasks.

Samad and Kanhere [34] proposed the Modified Reverse Auction (MRA), the winning probability of the participants is estimated and revealed individually before the auction is closed, and then, they are allowed to improve their winning probability by reducing their bid or increasing their contribution via moving to a different location. Zhou et al. [35] designed a novel delay-constraint and reverse auction-based incentive mechanism (DRAIM). In DRAIM, authors modeled the reverse auction-based incentive problem as a nonlinear integer problem, aiming to maximize the revenue and jointly consider the delay constraint in the optimization problem. To avoid malicious competition and select high-quality crowd workers to improve the utility of the crowdsourcing system, Hu et al. [36] proposed an incentive mechanism based on the combination of the reverse auction and multiattribute auction in mobile crowdsourcing. This mechanism adopts a dynamic threshold to make an online decision for whether to accept a crowd worker according to its attributes. To investigate the joint problem of sensing task assignment and schedule, Cai et al. [37, 38] proposed two distributed auction schemes (CPAS and TPAS). Luo et al. [39] designed an incentive mechanism for scenarios involving heterogeneous types of workers (and the beliefs about their respective types) using an asymmetric all-pay auction model.

In addition to the incentive mechanism based on auction theory, Wu et al. [40] designed a mechanism based on the Stackelberg game to encourage participants to compete and participate in tasks, where the requester determines a certain total payment amount from the beginning. Taking the prior knowledge as a specific incentive, Lan et al. [41] proposed a novel classifier that can accurately recognize different categories. An incentive mechanism for the platform-centric mobile crowdsourcing was designed by Zhan et al. [42], which considers the resource requirements of the users and resource constraints of smart devices. They formulated the interaction between the requester and workers as one-to-many bargaining; then, they studied the bargaining solutions under ordered bargaining and simultaneous bargaining systematically.

Although the above monetary incentive mechanisms are simple and effective, they also have some shortcomings. Due to the selfishness of individuals, their purpose is to maximize their benefits [43]. References [44, 45] pointed out that there are distrust problems such as free-riding problems and false-reporting problems. To encourage workers to choose a trust strategy, Wang et al. [44] proposed an online incentive mechanism based on a reputation for mobile crowdsourcing systems and established a reputation updating method. In order to prevent the free-riding problem of workers and motivate workers to contribute their efforts, Zhang et al. [45] proposed a novel class of incentive protocols based on social norms.

Most of the existing incentive mechanisms are short-term incentives that directly pay workers. These incentive mechanisms cannot attract users to participate in crowd tasks for a long time. When workers leave the crowdsourcing platform because they lose interest, the performance of the platform will significantly decrease. Gao et al. [46] proposed a Lyapunov-based Vickrey-Clarke-Groves auction policy for online sensor selection, which is aimed at maximizing social welfare and ensure the long-term participation incentive of users. Gao et al. [47] proposed a long-term quality perception incentive model in a crowdsourcing environment with budget constraints. The long-term incentive mechanism is to motivate crowd workers to provide long-term continuous services for crowd tasks. However, the above methods ignored the problem of workers' effort.

2.2. Repeated Game Theory. Inspired by game theory, the study of repeated games in incentive mechanisms has become a hot topic [48–51]. In repeated games, the player's goal changes from the current maximum profit to the maximum profit of multiple games, which means that future returns are closely related to the current behavior. Therefore, participants can be forced to avoid selfish behavior, and repeated games can also solve the dilemma. Mailath et al. [48] gathered and analyzed a metadata set of experiments on prisoners' dilemma games. They used experiments to prove that cooperation is affected by infinite repetition and that high cooperation rates are more likely to arise when it can be supported in equilibrium. To deal with the selfish behaviors of workers, Gao et al. [49] proposed an enhanced cooperative authentication protocol. For this designed protocol, an infinitely repeated game was proposed to analyze the utility of all users to help analyze the threat of selfish behavior. Hu et al. [50] proposed to model the interaction between the requester and crowd workers as a game process under the theoretical framework of repeated games; requesters adopted sequential zero-determination strategies to solve the crowdsourcing dilemma. Yin et al. [51] used a repeated game with incomplete information to motivate nodes to forward an advertisement.

2.3. Evolutionary Game Theory. In recent decades, the researches on evolutionary game theory (EGT) has become increasingly widespread [52]. Evolutionary stable strategy (ESS) and replication dynamics have been put forward successively in the course of its development, which laid the theoretical foundation for subsequent research. In EGT, if the later mutation behavior cannot shake a certain strategy executed by the population, that strategy is considered to be stable in the evolution process. ESS ensures stability and identifies robustness against mutations. Although the EGT was originally developed for biology, many exciting works have utilized EGT to model problems [53–55]. For example, Yin et al. [53] proposed an incentive mechanism based on EGT to inspire entities to select strategies that have high trustworthiness. To address the strategic uncertainty that users may face, EGT provided an excellent means. Wang et al. [54] used an evolutionary game framework to answer the question of "how to collaborate" in multiuser decentralized cooperative spectrum sensing. An evolutionary game is used to simulate the behavior of nodes in a network; Fang et al. [55] designed a budget allocation mechanism to encourage cooperation between adjacent nodes. The replicator dynamic can construct the gradual evolution of strategies to show the process of ESS. It is useful for investigating the trajectory of the strategies of players while adapting their

behaviors to reach the solution. Therefore, this paper combined the evolutionary game with the Wright-Fisher model for strategy evolution.

Therefore, according to the above problems, this paper proposed a multistrategy model to enhance the attractiveness of the platform under the framework of repeated games, which can not only ensure long-term participation of users but also guide workers to continuously improve their efforts.

3. System Overview

In this section, a generalized model in crowdsourcing systems is given, and then, the workflow of the crowdsourcing system is described in detail. Secondly, considering the historical behavior data of participants, according to different levels of effort, a multistrategy model is proposed.

3.1. System Description. A crowdsourcing system generally includes a crowdsourcing platform and users. Users could be divided into workers and requesters. Requesters could publish tasks through the crowdsourcing platform, and workers accept tasks and complete specific tasks with the help of mobile smart devices such as mobile phones. The crowdsourcing system could be elaborated from three aspects.

3.1.1. Requester. The requester publishes the task on the crowdsourcing platform; gives information about the task's time constraint, space constraint, and price; and then waits for feedback from the platform. The set of requesters is denoted by $R \triangleq \{1, 2, \cdots, j, \cdots, N\}$.

3.1.2. Worker. Workers could choose to complete tasks distributed by the platform, or they can choose tasks independently. To complete a specific task, workers need to provide specific data and information and then wait for feedback from the platform. The set of the workers is denoted by $W \triangleq \{1, 2, \cdots, i, \cdots, N\}$.

3.1.3. Platform. The crowdsourcing platform mainly includes two important parts, i.e., the tasks processing server and the payment server. The task processing server is responsible for task allocation, distributes tasks from requesters to appropriate workers, and promptly feeds back information about tasks required by both parties during the transaction. The payment server determines the worker's payment for this task based on the completion of the worker's task and distributes the payment to the designated worker.

According to the above three roles, the workflow of the crowdsourcing system includes the following steps. First of all, the requester releases the task information, the task processing server assigns the task, the worker chooses whether to accept the task, and if the task is accepted, the description of the available results will be returned to the task server. Then, the task server sends the task assignment result to the requester. Next, after receiving the result, the requester calculates the actual payment and sends the actual payment result feedback to the payment server. Finally, after receiving the information, the payment server informs the payment information to the worker. After the worker knows the actual payment, she submits the real data to the platform. The task server forwards the data to the requester, and the payment server sends the payment to the worker.

3.2. System Definition. According to the data contribution of workers, the efforts of workers are divided into different levels. The higher the level of effort workers provide, the greater the amount of contribution the requester will provide. Conversely, for workers with low effort levels who cannot meet the task requirements, the system will cancel their cooperation and achieve the purpose of punishment. According to different levels of effort, this paper adjusts the strategies of the participating parties in a targeted manner, to achieve different levels of income calculation functions.

When workers provide services, the level of effort of the worker $i, i \in W$, determines the amount of contribution Q_i the workers make in a specific task. In reality, workers can decide the contribution they want to provide. The relationship between the effort e of workers and the amount of contribution is shown by

$$Q_i(e_i) = e_i \times \varepsilon_i. \tag{1}$$

The random variable ε_i obeys a probability distribution function with an expectation of 0 and probability density function $f(\varepsilon)$.

Since the mobile devices used by workers in crowdsourcing consume resources such as power, memory, and time, a resource consumption cost $C(q_{ij})$ will be incurred when a certain amount of contribution is provided:

$$C\left(q_{ij}\right) = \frac{1}{2} \times \beta \times {q_{ij}}^2, \quad q_{ij} \in Q_i, \tag{2}$$

where q_{ij} represents the contribution provided by worker i to requester j; it belongs to the set of workers' contributions. $C(q_{ij})$ represents the negative utility to complete the task, *i.e.*, cost consumption, which is an increasing convex function, $C'(q_{ij}) > 0$ and $C''(q_{ij}) > 0$.

Definition 1 (utility of worker). Workers will have negative utility when they use mobile devices to provide data needed for tasks, and they will be paid after completing the tasks. The utility of workers is the difference between the payment and the negative utility:

$$U_i^j\left(q_{ij}\right) = P\left(q_{ij}\right) - C\left(q_{ij}\right), \tag{3}$$

where $U_i^j(q_{ij})$ represents the utility of the transaction between worker i and requester j. $P(q_{ij})$ means the payment to user i for data contribution q_{ij}.

Definition 2 (utility of requester). The information provided by the worker can bring a certain profit to the requester, and the requester needs to provide the worker with corresponding payment. The utility of the requester is the difference between the profit and the payment:

$$U_j^i\left(q_{ij}\right) = L\left(q_{ij}\right) - P\left(q_{ij}\right), \tag{4}$$

where U_j^i represents the utility of the transaction between worker i and requester j. $L(q_{ij})$ is the value of data contribution q_{ij}; the calculation method is shown in

$$L\left(q_{ij}\right) = \varsigma \times q_{ij}, \quad q_{ij} \in Q_i. \tag{5}$$

Definition 3 (effort level of worker). Although the requester cannot directly observe the real effort of the worker, he can know the amount of data contribution that the worker can provide. Therefore, the effort of the worker is evaluated according to the amount of data contribution that the worker can provide during the completion of a task. To ensure the quality of the collected data and ensure that workers have to work hard for each task, the requester can formulate a measurement method for the quality of the submitted data. This paper divides the interval according to the contributions submitted by the workers, which is calculated by

$$\text{Level} = \tanh\left(k \times \ln\left(\frac{q_{ij}}{q_{\text{normal}}}\right)\right), \tag{6}$$

where q_{normal} is the standard contribution of this task, i.e., the minimum contribution required to complete this task. The parameter k is an adjustable parameter. The reason for using these two functions is to quantify the data. When the level is less than 0, it indicates that the worker has not worked hard enough, and the data provided cannot meet the needs of the task.

In this paper, according to the actual data contribution of workers, the degree of effort is determined. The corresponding amount of data contribution provided by different degrees of effort is shown in Table 1.

Workers with different levels of effort have different utility functions. After analyzing Equation (6), it can be seen that the higher the level, the higher the level of effort of workers. Therefore, the utility calculation function of these five levels is obtained by

$$\begin{cases} U_{\text{level}_d}^i\left(q_{ij}\right) = (a + 1.5) \times P\left(q_{ij}\right) - C\left(q_{ij}\right), & \text{level}_d < 0, \\ U_{\text{level}_d}^i\left(q_{ij}\right) = P\left(b, q_{ij}\right) - C\left(q_{ij}\right), & \text{level}_d \geq 0,\ d = \{1, 2, 3, 4, 5\}, \end{cases} \tag{7}$$

where a and b are the left and right boundary values of different degree intervals, respectively.

4. Model and Formulation

In this section, dynamic games are used to model the interaction between workers and requesters; relevant rules of the game are given, and the existence of a dilemma is proposed. With the help of evolutionary games, it verifies that there are indeed dilemmas.

Table 1: Levels of effort.

Degrees of effort	Levels of effort
[-1, -0.5)	level_1
(-0.5, 0)	level_2
0	level_3
(0, 0.5)	level_4
(0.5, 1]	level_5

4.1. Game Formulation. In this section, the theoretical background of this model and related assumptions are given before modeling. In this model, the requester and the worker pursue profit maximization and individual utility maximization. Under the condition of information asymmetry, the requester needs to pay a certain amount to obtain information, which is to bring more benefits. Because the requester does not fully understand the worker's effort, the worker may be lazy. The task needs to be assigned to a series of workers who meet the requirements. It should be noted that from the requester's perspective, the actions of the workers required for a specific task follow the same pattern. Therefore, the success incentive for any worker means that it has a very high probability of success in this type of worker. This paper regards an interaction between the requester and any worker as a dynamic game process.

Specifically, a stage game is that after the requester publishes the task, the worker chooses the data contribution strategy to maximize her utility. Then, the requester decides whether to hire the worker by observing the strategy of the worker and decides how much to pay her. In this process, we assume the following:

Hypothesis 1: the platform has historical information about workers, and workers in the previous stage can participate in the tasks of the next stage after completing the tasks.

Hypothesis 2: ignore the cost consumption of supervision.

Hypothesis 3: the total duration of the game consists of a series of discrete stage games; each worker performs only one task in a stage.

Hypothesis 4: every game follows the same rules and processes.

Hypothesis 5: enough hard workers choose all tasks uniformly and randomly, and tasks have the same probability of being selected. This assumption does not reduce the number of potential workers. The mechanism proposed in this paper can actively guide workers to increase their efforts.

4.2. Game Model. In the stage game, the strategy of the worker is the amount of data contribution submitted to complete the task, while the strategy of the requester is whether to hire the worker. In this model, workers can use different contribution strategies. If the contribution is within the acceptance range of the requester, the requester chooses to hire workers and provide corresponding payment. The set of workers' strategies is defined as $(\text{level}_1, \text{level}_2, \text{level}_3, \text{level}_4, \text{and level}_5)$; the higher the level, the greater the contribution. The set of requesters' strategies is defined as (Y, N). Y and N

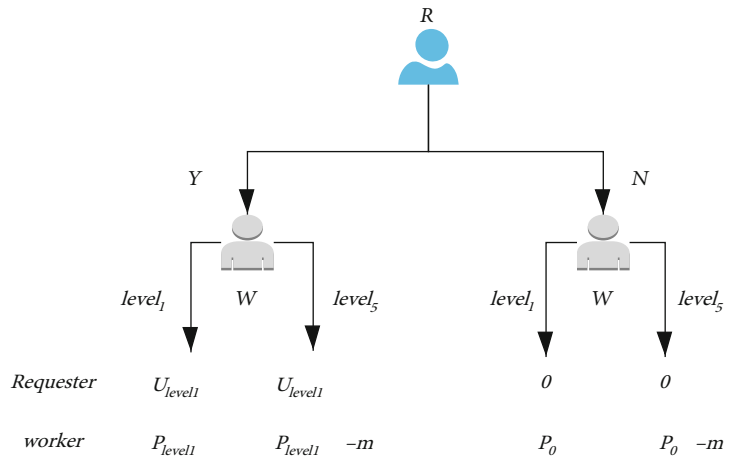


Figure 1: The game tree of a one-stage game between a requester and a worker.

indicate that the requester chooses to hire workers and the requester refuses to hire workers, respectively.

A stage game G is defined by $G = (N, S_k, U_k)$, $k \in N$, where $N = \{i, j\}$, $i \in W$, $j \in R$, is a finite set of players, S_k is the set of player k's actions, the strategy set of worker is $S_i = \{\text{level}_1, \text{level}_2, \text{level}_3, \text{level}_4, \text{level}_5\}$, the strategy set of the requester is $S_j = \{Y, N\}$, and U_k is player k's payoff function.

Definition 4 (Nash equilibrium solution of stage game). When S_i and Sj are fixed, if S_j^* satisfies $U_j(S_j{}^*) \geq U_j(S_j)$ and S_i^* satisfies $U_i^w(S_i{}^*) \geq U_i^w(S_i)$, we called (S_i^*, S_j^*) as the Nash equilibrium strategy in the proposed game.

Backward induction could be used to solve this Nash equilibrium. If the worker refuses this task, the worker becomes a self-employed person and only retains self-employment income p_0. Otherwise, workers choose their contribution strategy (level_1, level_2, level_3, level_4, and level_5). Providing data will bring negative utility to workers, and the greater the effort, the greater the negative utility. This paper first discusses the situation where workers have only two extreme strategies, level_1 and level_5, and extends it to a multi-strategy game. Through the game tree ($m > 0, U_{\text{level}_1} < 0$) shown in Figure 1, we can get the game result of backward induction. In the second step, workers will accept tasks and choose the level_1 strategy to reduce their negative utility. When the worker chooses the level_1 strategy, the requester will benefit more from the nonemployment strategy, so the requester will choose N. Therefore, the Nash equilibrium of this game is (level_1, N).

In this paper, we extend the two strategies to multiple strategies. Assuming that the requester hires worker i with a payment P^i, when the worker's effort is high, the benefits of the requester and the worker are the following: the requester's income is the difference between the profit obtained from the worker's data and the payment paid to worker i, and the worker's income is the difference between the payment from the requester and the negative utility. If the worker is lazy (that is, the effort level is level_1 or level_2), there is no negative effect.

Table 2: Benefit matrix of worker.

Strategy	Y	N
Level_1	$U_W(\text{level}_1, Y)$	$U_W(\text{level}_1, N)$
Level_2	$U_W(\text{level}_2, Y)$	$U_W(\text{level}_2, N)$
Level_3	$U_W(\text{level}_3, Y)$	$U_W(\text{level}_3, N)$
Level_4	$U_W(\text{level}_4, Y)$	$U_W(\text{level}_4, N)$
Level_5	$U_W(\text{level}_5, Y)$	$U_W(\text{level}_5, N)$

If the game is played only once at this stage, the benefits of both parties are shown in Tables 2 and 3; the calculation formulas are

$$\begin{cases} U_W(\text{level}_d, Y) = P^i, & d = \{1, 2\}, \\ U_W(\text{level}_d, Y) = P^i - C\left(q_{ij}\right), & d = \{3, 4, 5\}, \\ U_W(\text{level}_d, N) = P_0^i\left(q_{ij}\right), & d = \{1, 2, 3, 4, 5\}, \\ \text{s.t. } L\left(q_{ij}\right) - C\left(q_{ij}\right) > P_0 > \rho \times L\left(q_{ij}\right), \end{cases} \tag{8}$$

$$\begin{cases} U_R(\text{level}_d, Y) = \rho \times L\left(q_{ij}\right) - P^i, & d = \{1, 2\}, \\ U_R(\text{level}_d, Y) = L\left(q_{ij}\right) - P^i, & d = \{3, 4, 5\}, \\ U_R(\text{level}_d, N) = 0, & d = \{1, 2, 3, 4, 5\}, \\ \text{s.t. } L\left(q_{ij}\right) - C\left(q_{ij}\right) > P_0 > \rho \times L\left(q_{ij}\right). \end{cases} \tag{9}$$

After the above analysis, without any incentives, the result of the game is (level_1, N). This paper refers to this phenomenon as a crowdsourcing dilemma. Then, evolution games are used to evolve users' strategy changes. The existence of the crowdsourcing dilemma and the effectiveness

Table 3: Benefit matrix of requester.

Strategy	Y	N
Level_1	$U_R(\text{level}_1, Y)$	$U_R(\text{level}_1, N)$
Level_2	$U_R(\text{level}_2, Y)$	$U_R(\text{level}_2, N)$
Level_3	$U_R(\text{level}_3, Y)$	$U_R(\text{level}_3, N)$
Level_4	$U_R(\text{level}_4, Y)$	$U_R(\text{level}_4, N)$
Level_5	$U_R(\text{level}_5, Y)$	$U_R(\text{level}_5, N)$

of the incentive mechanism proposed in this paper have also been verified.

4.3. Evolution Analysis

4.3.1. Strategy Stability Analysis of Workers. Suppose n workers are participating in a task, and the number of workers in the five strategies are n_{level_d}, $d \in \{1, 2, 3, 4, 5\}$. The sum of the quantities is n. According to the income matrix, combined with the calculation method of the adaptability of different populations in the Wright-Fisher model, the adaptabilities of different strategies are calculated by

$$F_{\text{level}_d} = \frac{\left(n_{\text{level}_d} - 1\right) \times U_W(\text{level}_d, Y)}{n - 1}. \tag{10}$$

Since the Wright-Fisher process is to perform the N-fold Bernoulli experiment in the offspring set, the perceived worker obeys the binomial distribution. Assuming that n_i^m represents the number of workers at level i in the m th generation in the group, the probability that the number of workers at level i in the $m+1$th generation is n_i^{m+1} is shown by

$$p\left(n_{\text{level}_d}^{m+1} \middle| n_{\text{level}_d}^{m}\right) = \binom{n}{n_{\text{level}_d}^{m+1}} \prod_{d=1}^{5} \left(\frac{n_{\text{level}_d}^{m} \times F_{\text{level}_d}}{\sum_{d=1}^{5} n_{\text{level}_d}^{m} \times F_{\text{level}_d}}\right)^{n_{\text{level}_d}^{m+1}}. \tag{11}$$

In the Wright-Fisher process, individuals are updated synchronously; $E(\Delta x)/\Delta t$ in the game on the graph can be used to approximately replace the copy dynamic equation d_x/d_t in the evolutionary game. $E(\Delta x)$ represents the change in the individual frequency of the worker of strategy level_d; Δt indicates the compensation step of the update time. The calculation method $E(\Delta x)$ is shown by

$$\begin{aligned} E(\Delta x) &= \frac{\sum_{n_{\text{level}_d}^{m+1}=0}^{n} \left(n_{\text{level}_d}^{m+1} - n_{\text{level}_d}^{m}\right) \times p\left(n_{\text{level}_d}^{m+1} \middle| n_{\text{level}_d}^{m}\right)}{n} \\ &= \frac{n_{\text{level}_d}^{m} \times F_{\text{level}_d}}{\sum_{*}^{\text{level}_d} n_{*}^{m} \times F_{*}} - x. \end{aligned} \tag{12}$$

In this paper, the proportion of workers with five types of strategies is set x_d, $d \in \{1, 2, 3, 4, 5\}$, and the evolution prediction model of perceiving workers in the mobile crowdsourcing system can be obtained. Equation (13) is the evolution prediction model of the strategies:

$$\frac{dx_d}{dt} = \frac{x_d \times \overline{u_{\text{level}_d}}/\bar{u} - x_i}{\Delta t}. \tag{13}$$

The benefit calculation method is shown by Equations (14) and (15), where y means the proportion of requesters who decides to hire workers, $\overline{u_{\text{level}_d}}$ represents the expected benefits of workers at different levels of effort, and $\bar{u}$ represents the average expected benefits of all workers in the system:

$$\overline{u_{\text{level}_d}} = y \times U_W(\text{level}_d, Y) + (1 - y) \times U_W(\text{level}_d, N), \tag{14}$$

$$\bar{u} = \sum_{d=1}^{5} x_d \times \overline{u_{\text{level}_d}}. \tag{15}$$

According to the evolution prediction model, the evolution trend of different types of workers can be predicted. To find the ESS, that is, the Nash equilibrium, two conditions $F(x) = d_x/d_t = 0$ and $F(x)' < 0$ need to be met at the same time. The Nash equilibrium point is obtained through computing $x_d = 0$, $d \in \{2, 3, 4, 5\}$. It can be seen that due to the selfishness of the workers, the ESS of the workers is the strategy level_1, which also corresponds to the crowdsourcing dilemma in the previous section.

4.3.2. Strategy Stability Analysis of Requesters. When the worker's effort is low, the requester will adopt a N strategy. If this situation persists for a long time, the number of task transactions in the system will drop significantly, resulting in system performance degradation. To analyze the evolution trend of task requesters with different strategies, a trust evolution prediction model for task requesters is also established. Similarly, the requester's strategies include employment and nonemployment, the proportion of these two types of requesters is y_d, $d \in \{1, 2\}$, and the evolution prediction model of the two strategies is

$$\frac{dy_d}{dt} = \frac{y_d \times \overline{u_Y}/\bar{u} - y_d}{\Delta t}. \tag{16}$$

The calculation method of income is $\overline{u_Y}$ representing the expected return of the requester who chooses strategy Y. $\overline{u_N}$ indicates the expected return of the requester who chooses strategy N; $\bar{u}$ represents the average expected return of all requesters in the system:

$$\begin{aligned} \overline{u_Y} &= y_1 \times \left(\sum_{d=1}^{5} x_d \times U_R(Y, \text{level}_d)\right), \\ \overline{u_N} &= y_2 \times \left(\sum_{d=1}^{5} x_d \times U_R(N, \text{level}_d)\right), \\ \bar{u} &= y_1 \times \overline{u_Y} + y_2 \times \overline{u_N}. \end{aligned} \tag{17}$$

According to the evolution prediction model, the evolution trend of different types of requesters can be predicted. To find the ESS, that is, the Nash equilibrium, two conditions $F(y) = d_y/d_t = 0$ and $F(y)' < 0$ need to be met at the same time. The Nash equilibrium point is obtained through computing $y_1 = 0$. When the number of low-effort workers in the group is the majority, the requester's ESS in the system is not hired, which is consistent with the previous question of the crowdsourcing dilemma. To solve the above problems, a multistrategy repeated game incentive model was established to motivate workers. Requesters hire workers with appropriate remuneration and maintain the system to provide long-term and efficient services.

5. Design of Incentive Mechanism

To solve this dilemma, ensure the long-term participation of users and improve the quality of workers' completion. The interaction between workers and requesters is modeled as a repeated game, and a multistrategy incentive mechanism is proposed in this section.

5.1. Analysis of Multistrategy Repeated Game Model. In the repeated game, the game of the same structure is repeated many times, even infinite times. Players will adopt a confrontational strategy to maximize personal gains in a one-shot game. For example, the incident of a stranger grabbing a seat on a bus can be considered a game. If the two parties are not strangers, and they give up their immediate interests because of future interaction, there will be no quarrel. Therefore, repeated games can solve the dilemma problem. If it is repeated for a limited number of N times, then in the Nth game, the participants know that if they choose a low price in this game, they will benefit themselves and will not leave the opponent with any chance of revenge.

When the workers cannot predict when the game will end, the workers will not adopt a confrontation strategy in the last game; that is, there will be no critical state in which the worker's effort is low in the last game. Therefore, to eliminate the dilemma caused by selfishness, although personal working ability, time, and other conditions are limited conditions, the game between the two parties can still be regarded as an infinitely repeated game in the analysis process. Under a certain discount factor, the trigger strategy in the infinite repeat game can form a subgame Nash equilibrium, which can maintain a long-term high-level service.

The stage game is repeated continuously in time, and the historical information of the game can be obtained before the next game starts. Simply put, a repeated game is a stage game with the same structure repeated many times. Suppose G is a stage game. When G is repeated T times, it is called a repeated game $G(T)$; when T is limited, it is called a finite repeated game; and when T is infinite, it is called an infinite repeated game.

Definition 5 (infinite repeated game). For all $\delta \in [0, 1)$, the repeated game $G(T)$ consists of an infinite sequence of repetitions of G with a common discount factor δ. We denote that the definition of a repeated game is

$$G(T) = G(\infty) = (N, S_k, U_k, T, \delta), \quad k \in N, \quad T = 0, 1, 2, \cdots, t, \cdots, \infty. \tag{18}$$

In the repeated game $G(\infty)$, the strategy selected by the participants in stage t is defined as S_k^t, $S_k^t \in S_k$; then, the strategy combination in stage t is defined as $S^t = (S_1^t, S_2^t, \cdots, S_k^t)$. Supposed that the benefits of a certain player at each stage are $U_k(S^t)$; the present value of the total return in an infinitely repeated game is obtained by

$$\pi_k = \sum_{t=1}^{T} \delta^{t-1} U_k(s^t), \quad t \in T, \tag{19}$$

where $\delta \in [0, 1)$ is the discount factor, and the discount factor is the importance of future earnings in the current stage. The discount factor determines a credible threat. This threat makes no participant willing to violate the trigger strategy alone, which follows the principle of Nash equilibrium. The long-term utility is the normalized sum of the discounted expected stage utilities, which can also be called the average return. Assuming that the long-term utility of each stage is $\overline{\pi_k}$, the present value can be expressed as $\overline{\pi_k}/(1-\delta)$. According to Equation (19), the average return is obtained by

$$\overline{\pi_k} = (1-\delta) \sum_{t=1}^{\infty} \delta^{t-1} U_k(s^t), \quad t \in T. \tag{20}$$

Theorem 6 (folk theorem). *$G(T)$ is an infinite repeated game with G as the stage game, S^* is the Nash equilibrium of stage game G, $U_k(S^t)$ is a set of payment determined by S^*, and $U_k(S)$ is a set of viable payment. For any $k = 1, 2, \cdots, N$ satisfying $U_k(S^*) > U_k(S)$, there exists $\delta^* < 1$, so that all $\delta \geq \delta^*$, and S^* is a subgame perfect Nash equilibrium in each round of the game.*

The above theorem means that in an infinitely repeated game, if the participants have enough patience (the discount factor satisfies certain conditions), then any feasible payout vector that satisfies individual rationality can be obtained through a specific subgame Nash equilibrium.

5.2. Workflow of the Proposed Incentive Mechanism. In the proposed incentive mechanism, if a worker chooses a low-effort strategy in the tth transaction, she will never be able to participate in such crowdsourcing tasks after

completing this task. The crowdsourcing process with an embedded incentive mechanism is as follows. First of all, the requester publishes the task and the price of the task; then, the worker considers whether to accept the job or not. If she refuses the job, she only has the benefit of self-employment. To accept the job, an available data declaration needs to be submitted. Different contribution strategies have different negative utility, and higher effort will result in higher negative utility.

The requester considers whether to hire the worker for the task based on historical data. If the requester decides to hire the worker, the requester will calculate the corresponding payment after receiving the statement of the worker's contribution. The requester feeds back a series of decisions to the platform. The platform feeds back employment information to workers. Workers who are successfully employed submit real data and are paid when platform validation is passed. The trigger strategies of the two parties in the tth stage are the following.

5.2.1. Requester. If the effort level is greater than 0 in the previous t-1 stages and greater than 0 in the tth stage, then the worker will be hired and paid correspondingly in the tth stage; otherwise, it will not be hired.

5.2.2. Worker. Workers accept jobs when the price of the task is greater than the self-employment income. If the payment in the previous t-1 stages can compensate for the negative utility and there are additional benefits, the worker chooses to provide a high degree of effort.

The trigger strategy makes both parties have a threat. The threat of the requester is once the worker does not work hard, he will not hire the worker in the next stage. The threat of workers is if the payment is less than the self-employment income, they will not work hard. How much payment should the requester pay to the worker so that it is beneficial to the requester and meets the requirements of the worker? The following is a discussion of payment.

(1) First, from the perspective of workers, this paper assumes that the total income when workers do not violate the trigger strategy is π_e:

$$\begin{aligned}\pi_e &= \left(P\left(q_{ij}\right) - C\left(q_{ij}\right)\right) + \delta \times \left(P\left(q_{ij}\right) - C\left(q_{ij}\right)\right) \\ &\quad + \delta^2 \times \left(P\left(q_{ij}\right) - C\left(q_{ij}\right)\right) + \cdots + \delta^{t-1} \\ &\quad \times \left(P\left(q_{ij}\right) - C\left(q_{ij}\right)\right) + \cdots \\ &= \frac{1}{1-\delta} \times \left(P\left(q_{ij}\right) - C\left(q_{ij}\right)\right).\end{aligned} \tag{21}$$

Assuming that the worker chooses to violate the trigger strategy in the t period, ρ represents the probability of workers providing high contribution at the low effort level; the total income is

$$\begin{aligned}\pi_s &= \left(P\left(q_{ij}\right) - C\left(q_{ij}\right)\right) + \delta \times \left(P\left(q_{ij}\right) - C\left(q_{ij}\right)\right) + \cdots + \delta^{t-2} \\ &\quad \times \left(P\left(q_{ij}\right) - C\left(q_{ij}\right)\right) + \cdots + \delta^{t-1} \times P\left(q_{ij}\right) + \delta^t \\ &\quad \times \left(\rho \times P\left(q_{ij}\right) + P_0 \times (1-\rho)\right) + \delta^{t+1} \\ &\quad \times \left(\rho \times P\left(q_{ij}\right) + P_0 \times (1-\rho)\right) + \cdots.\end{aligned} \tag{22}$$

The sufficient condition to drive workers not to violate the trigger strategy is

$$\pi_e - \pi_s > 0. \tag{23}$$

Therefore, the constraint conditions for workers not to violate the trigger strategy are

$$P > P_0 + C\left(q_{ij}\right) + \frac{1-\delta}{\delta(1-\rho)} \times C\left(q_{ij}\right). \tag{24}$$

(2) Second, from the perspective of requesters, if the requester chooses not to hire workers, his profit is 0; therefore, the sufficient condition for the requester to follow the trigger strategy in the infinite repeat game is

$$\frac{1}{1-\delta}\left(L\left(q_{ij}\right) - P\left(q_{ij}\right)\right) > 0. \tag{25}$$

That is,

$$\left(L\left(q_{ij}\right) - P\left(q_{ij}\right)\right) > 0. \tag{26}$$

Through combining Equations (24) and (26), we get the final result:

$$L\left(q_{ij}\right) > P > P_0 + C\left(q_{ij}\right) + \frac{1-\delta}{\delta(1-\rho)} \times C\left(q_{ij}\right). \tag{27}$$

After the above analysis, inequality is obtained. The payment given under this constraint makes the trigger strategy of both parties constitute a subgame perfect Nash equilibrium. The calculation method of the discount factor in this paper is shown by

$$\delta \geq \delta^* = \frac{C\left(q_{ij}\right)}{C\left(q_{ij}\right) + (1-\rho)\left(L\left(q_{ij}\right) - P_0 - C\left(q_{ij}\right)\right)}. \tag{28}$$

Input: tasks set A
Output: A new set of tasks Γ with a discount factor
1: For $a \in A$ do
2: Suppose the triggering strategy is adopted in the first time of infinitely repeated game
3: calculate δ^* the according to Eq.(28)
4: Add the discount factor to the task set
5: End for

Algorithm 1: Discount factor.

Input: tasks set Γ**, workers set** W
Output: payoff set P_p
1: Initialize(P_p, σ)
2: for t =1 to T do
3: **For** $\gamma \in \Gamma$ do
4: Posts task information on the platform, including the price tag P_{tag} and contribution q_{normal}
5: Waiting for platform feedback
6: **for** $w \in$ W do
7: Calculate the degree of effort σ of worker w by Eq.(6)
8: Traverse worker's history information
9: **If** $\sigma(t) > 0$ and $\sigma(1,\cdots,t-1) > 0$ **then**
10: Decide to hire the worker and give feedback to the platform
11: Calculate payoff P_p(t) of worker by Eq.(29)
12: **else** Decide not to hire
13: **end if**
14: **end for**
15: Update (P_p⊠σ)
16: end for

Algorithm 2: MSRG for requesters.

Input :tasks set Γ**, workers set** W
Output : payoff set P_p
1: for t =1 to T do
2: **for** $\gamma \in \Gamma$ do
3: **If** $P_{tag} > P_0$ then
4: Submit the task contribution amount description to the platform
5: **If** worker are not hired then
6: Choose another task
7: **else**
8: Traverse worker's history information
9: **If** $P_p(1,\cdots,t-1) > P_0(1,\cdots,t-1) + C(1,\cdots,t-1)$ **then**
10: Choose high level ,and submit real data to the platform
11: **end if**
12: **end if**
13: **else** Choose another task
14: **end if**
15: **end for**
16: end for

Algorithm 3: MSRG for workers.

Based on the above analysis, the calculation method of workers' payment is given, where b means the right boundary of different effort levels:

$$P = P_0 + C\left(q_{ij}\right) + \frac{1-\delta}{\delta(1-\rho)} \times C\left(q_{ij}\right) \times (1+b). \quad (29)$$

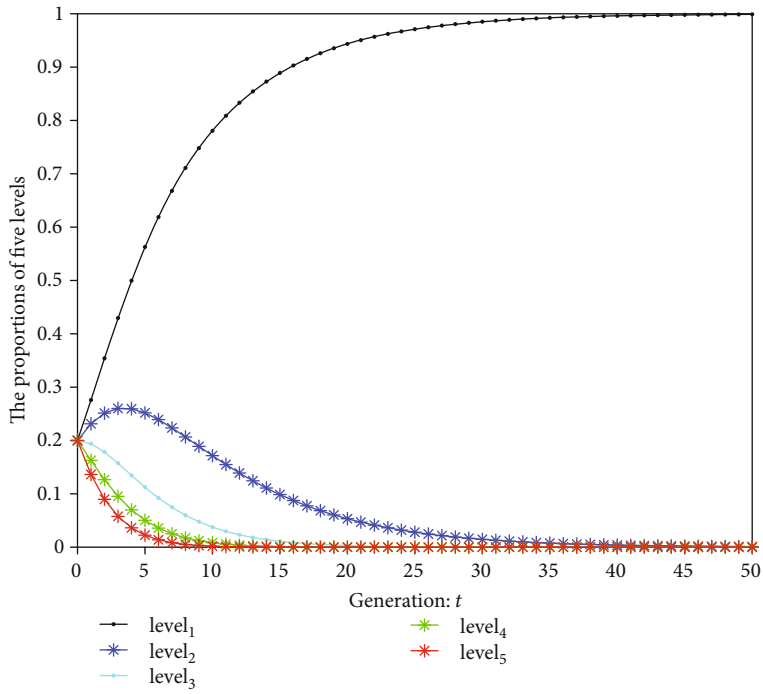


Figure 2: The evolution of the worker's strategy under the crowdsourcing dilemma.

Based on the above analysis, this paper proposes a multistrategy repeated game algorithm—the MSRG algorithm. Algorithms can be implemented in a decentralized manner, which are shown by Algorithms 2 and 3. First, Algorithm 1 is used to process the task set to get the discount factor. Then, from lines 6 to 15 in Algorithm 2, the requester calculates the effort of the workers based on the amount of contribution and decides whether to hire workers and how much they are paid based on historical information. The workers from lines 3 to 12 in Algorithm 3 decide whether to accept the job and determine the degree of effort for this task based on historical information.

6. Simulation Experiment Analysis

In order to verify the effectiveness of the multistrategy repeated game incentive mechanism proposed in this paper, four sets of simulation experiments are designed. The first set of experiments verifies the existing crowdsourcing dilemma. The second set of experiments verifies the optimal discount factor in the range of discount factor satisfying equilibrium. The third set of experiments verifies that the incentive mechanism can effectively solve the existing crowdsourcing dilemma. The fourth set of experiments compares the MSRG incentive mechanism with the simple crowdsourcing mechanism (general crowdsourcing mechanism without incentive mechanism) to illustrate the effectiveness of multistrategy repeated games.

The simulation experiment environment in this paper is the Windows 10 operating system, Intel® Core™ i5-6500 CPU @3.20 GHz 8 GB memory, Matlab2018a, and JetBrains PyCharm 2018.3.3 simulation platform. We simulate the algorithm by setting $\beta = 0.1$, $k = 2$, $q_{\text{normal}} = 8$, and $\varsigma = 1$. Other parameter settings will be introduced separately in each group of simulation experiments.

6.1. Verify the Existence of the Crowdsourcing Dilemma. According to the game payoff matrixes in Tables 2 and 3, we conduct the experiments to verify the crowdsourcing dilemma phenomenon in mobile crowdsourcing systems. In the experiments, we set the initial value of the workers with five levels of effort to account for 0.2, 0.2, 0.2, 0.2, and 0.2, respectively. It can be seen from Figure 2 that the four strategies except level_1 have a short rise, but they all converge to 0 in the end. After 23 iterations, level_1 finally converges to 1. Therefore, the experimental results verify that the workers in the crowdsourcing dilemma have a low level of effort due to selfishness.

Figure 3 presents the evolution of the requester's strategy under the crowdsourcing dilemma, with different worker settings. The proportions of low-effort workers in the experimental setup system are 0.8, 0.7, and 0.4, respectively. One can see that when there are more low-effort workers, no matter what the initial value of the requester who chooses Y, the final strategy of the requester is not to hire. This is because the data provided by the workers cannot meet the needs of the task and cannot bring ideal benefits to the requester. Therefore, the best strategy for the requester is not to hire. This result obviously justifies the dilemma raised in crowdsourcing.

6.2. Optimal Discount Factor. If the discount factor is too large, Equation (27) shows that the current pay is low, which will make the current task less attractive to workers. Therefore, when the discount factor is too large, workers cannot be guided to a high level of effort. If the discount factor is

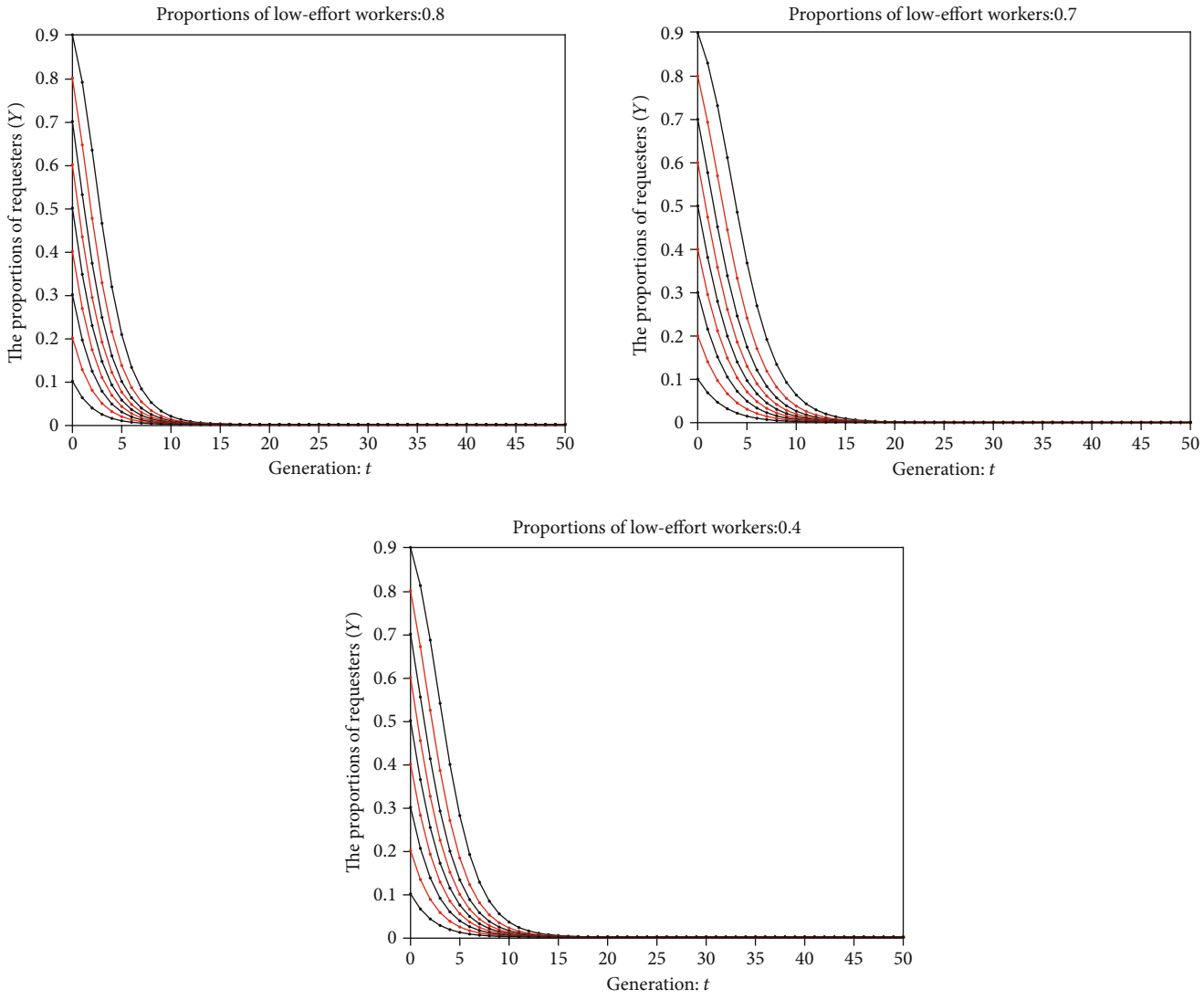


Figure 3: The evolution of the requester's strategy under the crowdsourcing dilemma.

too small, or even less than δ^*, then the triggering strategy cannot constitute a perfect Nash equilibrium of subgame, resulting in the failure to solve the crowdsourcing dilemma. In addition, the discount factor is too small to cause workers to not pay attention to future tasks, thus cannot maintain long-term incentive. In order to verify the influence of different discount factors on system performance, the range of discount factors is obtained according to Equation (28). The experiments are carried out to find the optimal discount factor.

The parameter δ^* is set to be 0.7, 0.8, and 0.9, respectively. The experimental results are shown in Figure 4; it shows that when δ^* is 0.7, the population of level_5 converges to 1 fastest. According to Equation (27), it is found that the smaller the discount factor, the higher the reward. Therefore, within the range of satisfying the equilibrium, the smaller the discount factor, the greater the benefits of workers. That is to say, when the discount factor is 0.7, the convergence rate is the fastest.

Figure 5 shows the evolution of the requester strategy under different discount factors as the number of evolutionary generations increases. It can be seen from Figure 4 that when the discount factor is 0.5 and 0.3, the final strategy of the requester is N. This is because the excessively high reward cost cannot bring the desired benefits to the requester, so strategy N is the optimal choice for the requester. From Equation (27), it is obvious that the smaller the discount factor, the higher the current reward given by the requester. It will cause workers to no longer value long term, resulting in an increasing proportion of low-quality workers in the system, which will also cause the requester to eventually converge to strategy N.

6.3. Effectiveness of Incentive Mechanism. In the experiments, we set the initial values of the five worker strategies to account for 0.2, 0.2, 0.2, 0.2, and 0.2, respectively. The discount factor is set to be 0.7. The evolution of the worker

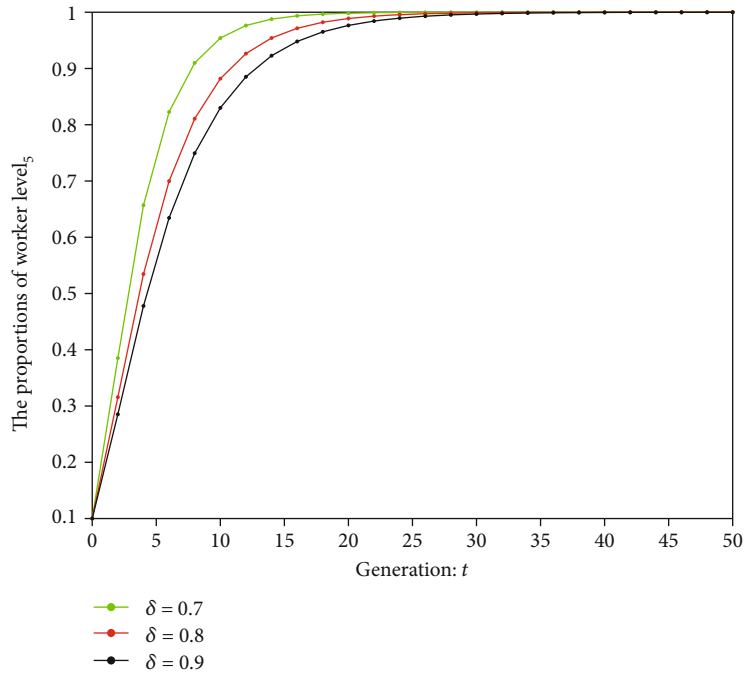


Figure 4: The evolution of the worker's strategy under different discount factors.

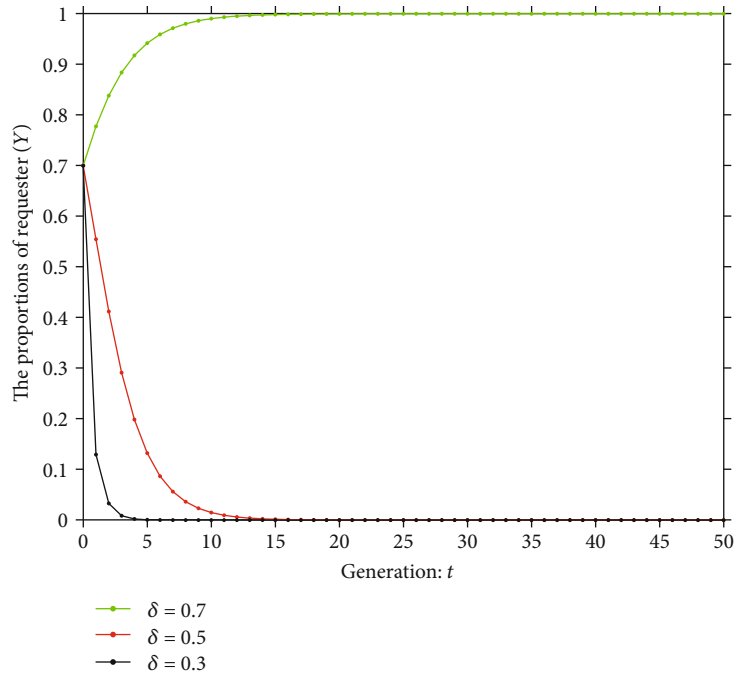


Figure 5: The evolution of the requester's strategy under different discount factors.

and requester under the MSRG incentive mechanism is presented in Figures 6 and 7.

In Figure 6, one can clearly observe that when incentives are added, the worker population will eventually adopt the level$_5$ strategy, which shows that the crowdsourcing dilemma has been resolved. This is because the higher the worker's level of effort, the higher the task's payment benchmark and rewards they will receive. Out of individual rationality, workers will adopt the level$_5$ strategy if they want to maximize their own benefits.

The level of worker's effort with the incentive mechanism increases continuously, and the change of the requester's strategy is shown in Figure 7. Figure 7 shows

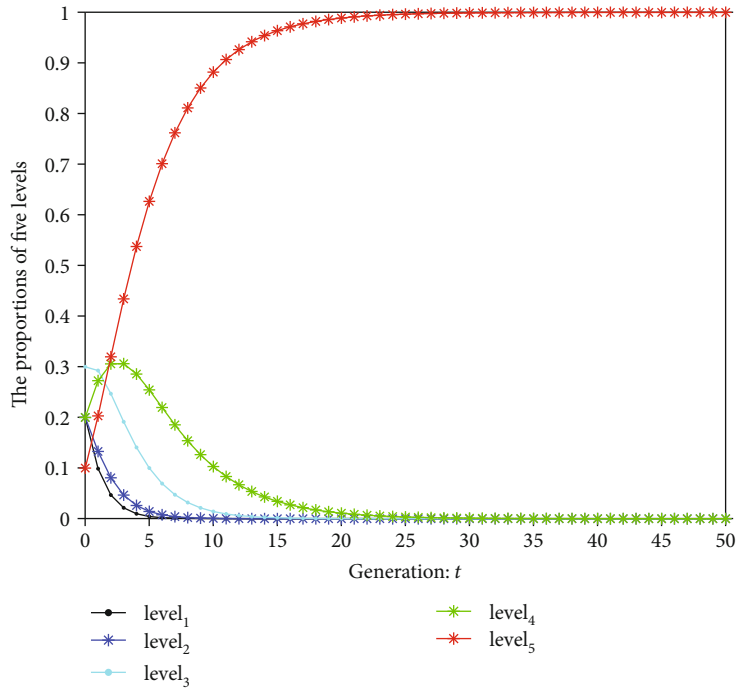


Figure 6: The evolution of worker's strategies under MSRG incentive mechanism.

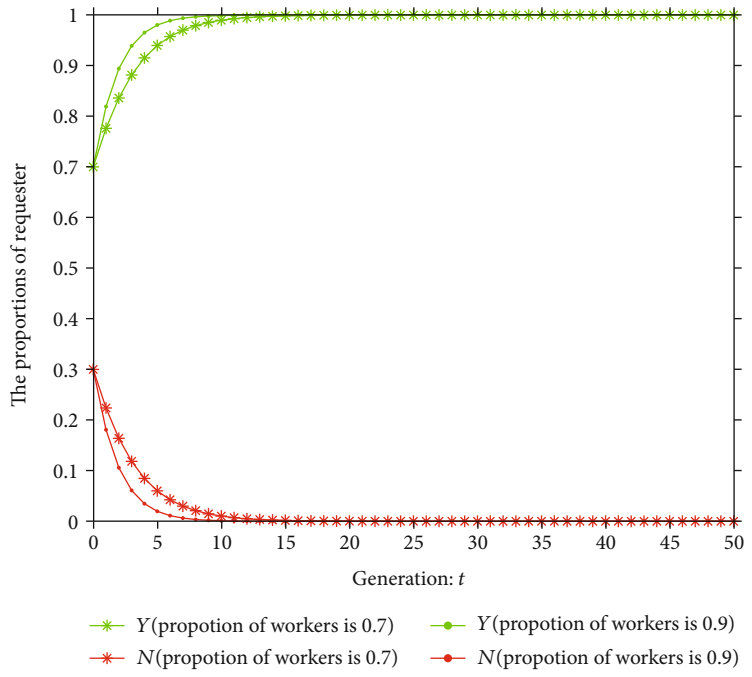


Figure 7: The evolution of the requester's strategy under MSRG incentive mechanism.

the changes of the requester's strategies when the proportion of high-effort workers is 0.7 and 0.9, respectively. It can be seen that the higher the proportion of workers with a high-effort level, the faster the requester's convergence to the employment strategy, and no matter what the initial value of the requester is, it will eventually converge to employment. This is because with the MSRG incentive mechanism, the data contribution of workers is kept at a high level, so the requester will gain more by employing the strategy.

6.4. Budget and Contribution. The experiment set the proportion of workers of each level in the initial group as 0.2, 0.2, 0.2, 0.2, and 0.2. $\ln(t)$ is a function related to the stage t,

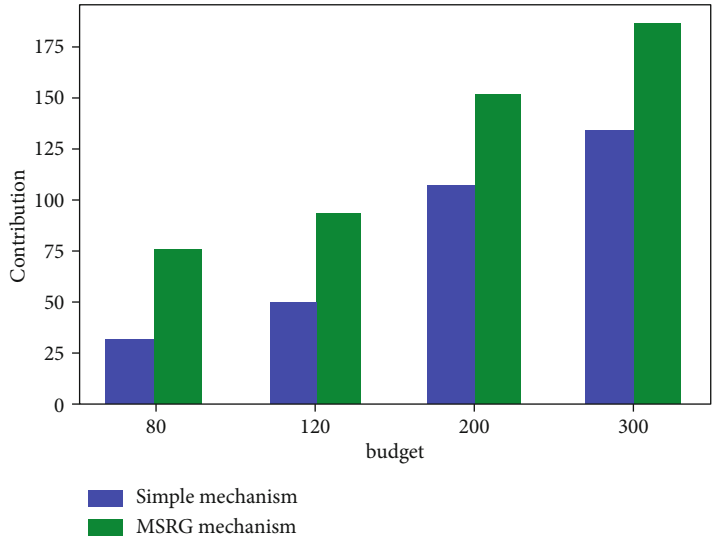


Figure 8: Comparison of the contributions of the two mechanisms.

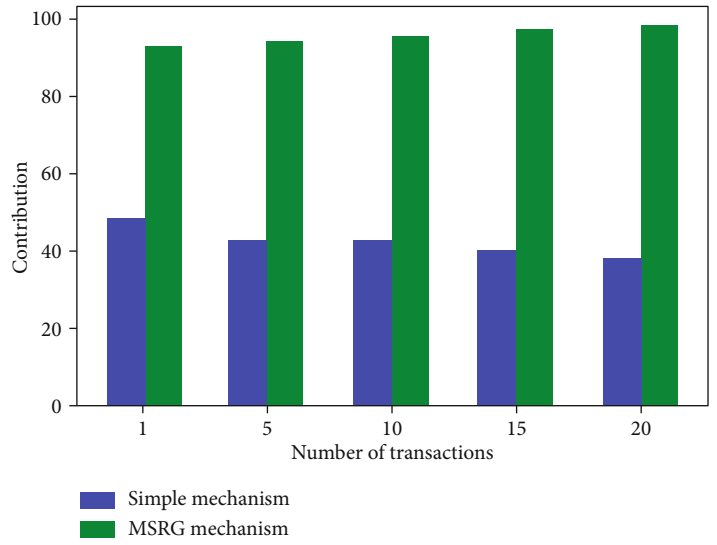


Figure 9: Comparison of the contributions of the two mechanisms.

which can be used as an adjustment parameter of the worker's strategy. The contributions of the five levels were $\ln(t) + 3, \ln(t) + 5, \ln(t) + 8, \ln(t) + 8.5$, and $\ln(t) + 9$. According to the evolutionary rules in this mechanism, the number of high-level workers is constantly increasing. The evolution of the worker strategy follows the evolution rules proposed by the MSRG mechanism.

Figure 8 shows the contribution of the two mechanisms under different budgets. It can be seen that under the same budget, requesters in the MSRG incentive mechanism can get more data contributions. This is because the MSRG mechanism can make judgments on the level of workers' effort, thus avoiding the hiring of low-level workers, which means that the payment will not be paid to workers with low contributions.

Assuming a task's budget is 120, the total contributions of the 1st, 5th, 10th, 15th, and 20th tasks are computed in the experiment. Figure 9 describes the amount of data contribution for the same budget in different trading periods. As the number of transactions increases, the amount of contribution received by the requester also increases and remains at a high level after 20 transactions. This is because after 20 transactions have been conducted, the proportion of workers with the highest level of effort in the group could reach 1.

As the number of task transactions increases, the advantages of this mechanism become more obvious. In Figure 10, one can see that as the number of transactions increases, the difference in contribution between the two mechanisms gradually increases. It is obvious that workers need to continuously improve their level of effort to get a higher payment.

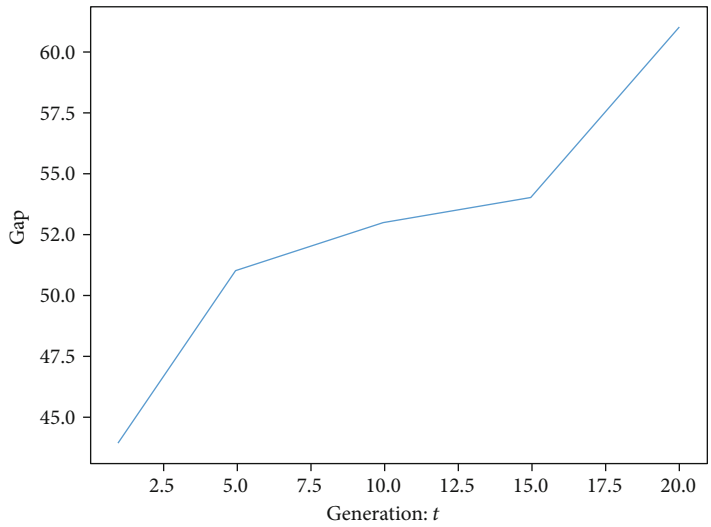


Figure 10: Change of the contribution gap.

Therefore, the contribution provided by the workers in the system remains at a high level. This experiment can prove that the MSRG incentive mechanism could maintain the long term and inspire workers to provide high-quality sensing data.

7. Conclusion

In this paper, we propose an incentive mechanism based on multistrategy repeated games to encourage long-term participation of workers. The discount factor and historical data of repeated games are used to obtain the current optimal behaviors of both sides in the game, and the MSRG algorithm is proposed to obtain the Nash equilibrium and find the optimal return. When the crowdsourcing dilemma is resolved, the long-term incentive of the system is also guaranteed. This paper also uses an evolutionary game to verify the effectiveness of this mechanism. The proposed MSRG incentive mechanism could guide workers to continuously improve their level of effort through monetary incentives. The extensive simulation experiments demonstrate that our proposed algorithms can guide workers to maintain a high level of effort and maintain the long-term incentive of mobile crowdsourcing systems.

In future works, we will further investigate other issues related to service quality between transactions in MEC, such as privacy protection issues. Based on game theory, we further improve the incentive mechanism to solve the contradiction between privacy protection loss and service quality [56].

Data Availability

In order to verify the effectiveness of the multistrategy repeated game incentive mechanism proposed in this paper, four sets of simulation experiments are designed. Specific experimental parameter settings can be found in Simulation Experiment Analysis.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62072392, Grant 61822602, Grant 61772207, Grant 61802331, Grant 61602399, Grant 61702439, Grant 61773331, and Grant 62062034; the China Postdoctoral Science Foundation under Grant 2019T120732 and Grant 2017M622691; the Natural Science Foundation of Shandong Province under Grant ZR2016FM42; the Major scientific and technological innovation projects of Shandong Province under Grant 2019JZZY020131; and the Key projects of Shandong Natural Science Foundation under Grant NO. ZR2020KF019.

References

[1] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[2] L. Qi, C. Hu, X. Zhuang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[3] C. Chen, Y. Zhang, M. R. Khosravi, Q. Pei, and S. Wan, "An intelligent platooning algorithm for sustainable transportation systems in smart cities," *IEEE Sensors Journal*, p. 1, 2020.

[4] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

[5] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven web APIs recommendation for building web applications," *IEEE Transactions on Big Data*, p. 1, 2020.

[6] Q. Duan, S. Wang, and N. Ansari, "Convergence of networking and cloud/edge computing: status, challenges, and

opportunities," *IEEE Network*, vol. 34, no. 6, pp. 148–155, 2020.

[7] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems," *Computer Networks*, vol. 135, pp. 32–43, 2018.

[8] https://answers.yahoo.com/,2020.

[9] H. Hong, X. Li, D. He, Y. Zhang, and M. Wang, "Crowdsourcing incentives for multi-hop urban parcel delivery network," *IEEE Access*, vol. 7, pp. 26268–26277, 2019.

[10] S. Zhu, Z. Cai, H. Hu, Y. Li, and W. Li, "zkCrowd: a hybrid blockchain-based crowdsourcing platform," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4196–4205, 2020.

[11] J. Li, Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, 2016.

[12] T. Liu, Y. Wang, Y. Li, X. Tong, L. Qi, and N. Jiang, "Privacy protection based on stream cipher for spatiotemporal data in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7928–7940, 2020.

[13] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Computer Networks*, vol. 102, pp. 157–171, 2016.

[14] Y. Wu, J. Zeng, H. Peng, H. Chen, and C. Li, "Survey on incentive mechanisms for crowd sensing," *Journal of Software*, vol. 27, pp. 2025–2047, 2016.

[15] J. Li, Z. Cai, J. Wang, M. Han, and Y. Li, "Truthful incentive mechanisms for geographical position conflicting mobile crowdsensing systems," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 324–334, 2018.

[16] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[17] T. Shi, Z. Cai, J. Li, and H. Gao, "CROSS: a crowdsourcing based sub-servers selection framework in D2D enhanced MEC architecture," in *The 40th IEEE International Conference on Distributed Computing Systems (ICDCS 2020)*, Singapore, 2020.

[18] J. Lu, Y. Xin, Z. Zhang, S. Tang, C. Tang, and S. Wan, "Extortion and cooperation in rating protocol design for competitive crowdsourcing," *IEEE Transactions on Computational Social Systems*, p. 1, 2020.

[19] L. Zhang, Y. Ding, X. Wang, and L. Guo, "Conflict-aware participant recruitment for mobile crowdsensing," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 192–204, 2020.

[20] A. Wang, M. Ren, H. Ma, L. Zhang, P. Li, and L. Guo, "Maximizing user type diversity for task assignment in crowdsourcing," *Journal of Combinatorial Optimization*, vol. 40, no. 4, pp. 1092–1120, 2020.

[21] Y. Wang, G. Yin, Z. Cai, Y. Dong, and H. Dong, "A trust-based probabilistic recommendation model for social networks," *Journal of Network and Computer Applications*, vol. 55, pp. 59–67, 2015.

[22] S. K. Goudos, Z. D. Zaharis, and K. B. Baltzis, "Particle swarm optimization as applied to electromagnetic design problems," *International Journal of Swarm Intelligence Research*, vol. 9, no. 2, pp. 47–82, 2018.

[23] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, p. 106196, 2020.

[24] Z. Cai and T. Shi, "Distributed query processing in the edge assisted IoT data monitoring system," *IEEE Internet of Things Journal*, p. 1, 2020.

[25] Y. Zhang, H. Qin, B. Li, J. Wang, S. Lee, and Z. Huang, "Truthful mechanism for crowdsourcing task assignment," *Tsinghua Science and Technology*, vol. 23, no. 6, pp. 645–659, 2018.

[26] Y. Wang, Y. Gao, Y. Li, and X. Tong, "A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems," *Computer Networks*, vol. 171, p. 107144, 2020.

[27] J. Xu, S. Wang, N. Zhang, F. Yang, and X. Shen, "Reward or penalty: aligning incentives of stakeholders in crowdsourcing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 974–985, 2019.

[28] J. Lu, Z. Zhang, J. Wang, R. Li, and S. Wan, *A Green Stackelberg-Game Incentive Mechanism for Multi-Service Exchange in Mobile Crowdsensing*, ACM ToIT, 2020.

[29] J. Liu, W. Wang, D. Li, S. Wan, and H. Liu, "Role of gifts in decision making: an endowment effect incentive mechanism for offloading in the IoV," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6933–6951, 2019.

[30] Y. Wang, Z. Cai, Z. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 1033–1046, 2020.

[31] J. Xu, S. Wang, B. K. Bhargava, and F. Yang, "A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3538–3547, 2019.

[32] Z. Duan, W. Li, X. Zheng, and Z. Cai, "Mutual-preference driven truthful auction mechanism in mobile crowdsensing," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1233–1242, Dallas, TX, USA, 2019.

[33] Y. Wang, Z. Cai, Z. Zhan, Y. Gong, and X. Tong, "An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 414–429, 2019.

[34] S. Samad and S. S. Kanhere, "MRA: a modified reverse auction based framework for incentive mechanisms in mobile crowdsensing systems," *Computer Communications*, vol. 145, pp. 137–145, 2019.

[35] H. Zhou, X. Chen, S. He, J. Chen, and J. Wu, "DRAIM: a novel delay-constraint and reverse auction-based incentive mechanism for WiFi offloading," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 4, pp. 711–722, 2020.

[36] Y. Hu, Y. Wang, Y. Li, and X. Tong, "An incentive mechanism in mobile crowdsourcing based on multi-attribute reverse auctions," *Sensors*, vol. 18, no. 10, p. 3453, 2018.

[37] Z. Cai, Z. Duan, and W. Li, "Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[38] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems,"

in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 635–644, Atlanta, GA, 2017.

[39] T. Luo, S. S. Kanhere, S. K. Das, and H. Tan, "Incentive mechanism design for heterogeneous crowdsourcing using all-pay contests," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2234–2246, 2016.

[40] W. Wu, W. Wang, M. Li et al., "Incentive mechanism design to meet task criteria in crowdsourcing: how to determine your budget," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 502–516, 2017.

[41] R. Lan, Y. Zhou, Z. Liu, and X. Luo, "Prior knowledge-based probabilistic collaborative representation for visual recognition," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1498–1508, 2020.

[42] Y. Zhan, Y. Xia, and J. Zhang, "Incentive mechanism in platform-centric mobile crowdsensing: a one-to-many bargaining approach," *Computer Networks*, vol. 132, pp. 40–52, 2017.

[43] Y. Zhang and D. S. M. Van, "Reputation-based incentive protocols in crowdsourcing applications," in *2012 Proceedings IEEE INFOCOM*, pp. 2140–2148, Orlando, FL, 2012.

[44] Y. Wang, Y. Li, Z. Chi, and X. Tong, "The truthful evolution and incentive for large-scale mobile crowd sensing networks," *IEEE Access*, vol. 6, pp. 51187–51199, 2018.

[45] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Keep your promise: mechanism design against free-riding and false-reporting in crowdsourcing," *IEEE Internet of Things Journal*, vol. 2, pp. 562–572, 2017.

[46] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2803–2811, Kowloon, 2015.

[47] L. P. Gao, T. Jin, and C. Lu, "A long-term quality perception incentive strategy for crowdsourcing environments with budget constraints," *International Journal of Cooperative Information Systems*, vol. 29, pp. 901–914, 2020.

[48] G. J. Mailath, I. Obara, and T. Sekiguchi, "The maximum efficient equilibrium payoff in the repeated prisoners' dilemma," *Games and Economic Behavior*, vol. 40, pp. 99–122, 2001.

[49] L. Gao, N. Ruan, and H. Zhu, "Efficient and secure message authentication in cooperative driving: a game-theoretic approach," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, 2016.

[50] Q. Hu, S. Wang, P. Ma, X. Cheng, W. Lv, and R. Bie, "Quality control in crowdsourcing using sequential zero-determinant strategies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 998–1009, 2020.

[51] L. Yin, Y. Guo, F. Li, Y. Sun, J. Qian, and A. Vasilakos, "A game-theoretic approach to advertisement dissemination in ephemeral networks," *World Wide Web*, vol. 21, no. 2, pp. 241–260, 2018.

[52] X. Liu, B. Cheng, and S. Wang, "Availability-aware and energy-efficient virtual cluster allocation based on multi-objective optimization in cloud datacenters," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 972–985, 2020.

[53] G. Yin, Y. Wang, Y. Dong, and H. Dong, "Wright-Fisher multi-strategy trust evolution model with white noise for Internetware," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7367–7380, 2013.

[54] B. Wang, K. J. R. Liu, and T. C. Clancy, "Evolutionary cooperative spectrum sensing game: how to collaborate?," *IEEE Transactions on Communications*, vol. 58, no. 3, pp. 890–900, 2010.

[55] L. Fang, G. Shi, L. Wang, Y. Li, S. Xu, and Y. Guo, "Incentive mechanism for cooperative authentication: an evolutionary game approach," *Information Sciences*, vol. 527, pp. 369–381, 2020.

[56] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8899186, 31 pages
https://doi.org/10.1155/2021/8899186

Research Article

Dispersed Computing for Tactical Edge in Future Wars: Vision, Architecture, and Challenges

Haigen Yang,[1] **Gang Li**,[1] **GuiYing Sun**,[2] **JinXiang Chen**,[3] **Xiangxin Meng**,[4] **HongYan Yu**,[4] **Wenting Xu**,[4] **Qiang Qu**,[5] **and Xiaokun Ying**[5]

[1]*Engineering Research Center of Wider and Wireless Communication Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 21003, China*
[2]*Chinese People's Liberation Army No. 61416, Beijing 100089, China*
[3]*Beijing Xinli Machinery Limited Liability Company, Beijing 100010, China*
[4]*Beijing Electro-Mechanical Engineering Institute, Beijing 100074, China*
[5]*China North Industry Advanced Technology Generalization Institute, Beijing 100089, China*

Correspondence should be addressed to Haigen Yang; yhg@njupt.edu.cn

Received 5 August 2020; Revised 6 December 2020; Accepted 17 December 2020; Published 7 January 2021

Academic Editor: Shaohua Wan

In the future, the tactical edge is far away from the command center, the resources of communication and computing are limited, and the battlefield situation is changing rapidly, which leads to the weak connection and fast changes of network topology in a harsh and complex battlefield environment. Thus, to meet the needs of communication and computing to build a new generation of computing architecture for real-time sharing and service collaboration of tactical edge resources to win the future war, the dispersed computing (DCOMP) seeks a new solution to satisfy the requirements of fast and efficient sensing, transmission, integrating, scheduling, and processing of various information in the tactical edge. Through the research of a traditional computing paradigm of mobile cloud computing (MCC), fog computing (FC), mobile edge computing (MEC), mobile ad hoc network (MANET), etc., it can be found that these computations have difficulty in meeting the high changing and complex battlefield environment and we propose a novel architecture of DCOMP to build a scalable, extensible, and robust decision-making system, to realize powerful and secure communication, computing, storage, and information processing capabilities for the tactical edge. We illustrate the fundamental principles of building a network model, channel allocation, and forwarding control mechanism of the network architecture for DCOMP called DANET and then design a new architecture, programming model, task awareness, and computing scheduling for DCOMP. Finally, we discuss the main requirements and challenges of DCOMP in future wars.

1. Introduction

With the evolution of the pattern of the major power relationships and the development of military science and technology, the military action may be far away from the homeland in the future called the tactical edge environment. These actions at the tactical edge usually lack support of communication and data processing capabilities, also known as the "first tactical mile," which is far away from the command center, with limited resources of communication and computing. The battle rhythm changes unexpectedly, leading to the frequent fluctuations of network connectivity and rapid changes of topology because of the highly dynamic and complex battlefield environment. These future military actions are a typical "uncertain fog of war" in the tactical edge, which is difficult to directly use the computing infrastructure at the command center in such harsh and complex war environment. It is extremely important to design and establish a new network and computing architecture by using advanced communication and computing technology oriented to the tactical edge, to realize the rapid perception of the situation of the battlefield, the quick integration and scheduling of the resources between the tactical edge nodes, and the efficient processing and transmission of battlefield information.

In recent years, the U.S. Army has always been at the forefront of leading military information technologies and proposed the concept of "net-centric operations and warfare," which is combined with the key role of "information superiority" and "decision superiority." In September 1999, the U.S. Department of Defense (DOD) proposed the concept of "global information grid (GIG)," which represents the third-generation development direction of the Internet [1]. In 2013, the U.S. Air Force proposed the concept of "combat cloud," which integrates the tactical communication network to realize quick exchange of data and resources of each combat unit in the Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) system [2]. In 2014, the Office of Naval Research (ONR) of U.S. proposed the concept of "tactical cloud," to achieve the data and applications of real-time awareness and process in the battlefield [3]. The "tactical cloud" needs to solve the problems of maintaining information consistency in the condition of the "high dynamic, weak connection" communication environment, software and data security in the cloud environment, dynamic "application tailoring" in different physical platforms, and real-time or near real-time processing of tactical data. In 2016, the Defense Advanced Research Projects Agency (DARPA) proposed a novel Internet architecture: "Dispersed Computing (DCOMP)" program, which is considered the next-generation battlefield environment support technology for the U.S. Army.

In June 2016, Information Innovation Office of DARPA released the proposals of an innovative research soliciting project for DCOMP [4], which is aimed at producing software instantiations of algorithms and protocol stacks that leverage pervasive, physically DCOMP platforms to boost application and network performance by orders of magnitude. The program is comprised of the three technical areas (TAs) as follows and described in more detail below:

(i) *TA1*. Algorithms for dispersed task-aware computation is aimed at developing algorithms and control mechanisms to enable efficient use of networked, geographically dispersed, heterogeneous computing capabilities in a manner consistent with the user, application, and task requirements

(ii) *TA2*. Programmable nodes and protocol stacks want to demonstrate the unique value that accrues from the presence of the programmable protocol logic within the network, primarily at the transport and application layers (but also potentially at the network layer) of the five-layer protocol stack model. TA2 systems may include new functions on users' terminal devices that interact with networked computation points (NCPs) in the network to optimize the overall performance

(iii) *TA3*. Technology integration describes how to combine itself with potential technologies that implement concepts across TA1 and TA2

DARPA entrusted Raytheon BBN Technologies (the premier research and development centers of Raytheon), Vencore Inc., BAE Systems Inc., and LGS Innovations to implement the DCOMP plan. The purpose of the DCOMP project is to improve the ability of task data calculation by using local computing resources, improve the reliability of the field network, distinguish available computing resources, specify the computing tasks of data in order of importance, and try to redesign and innovate the network protocol of the traditional Internet architecture, so as to significantly reduce the delay and bandwidth consumption and improve the performance of applications in a complex and uncertain battlefield environment [5].

The works and contributions of this article are listed as follows:

(i) Based on the studies of DCOMP, we propose the network model, channel allocation, and forwarding control mechanism of the DCOMP called DANET to achieve in realizing a centerless, multihop, self-organizing, infrastructure-free tactical edge network

(ii) We propose the architecture, software stack, programmable model, programmable language, programmable network, task awareness, and computing scheduling for DCOMP

(iii) Furthermore, we illustrate that the advantages, application scenarios, and the simulation result of the improved routing protocol for DCOMP are given

2. Related Work

Recently, the concepts such as distributed computing, cloud computing, MCC, cloudlet, FC, EC, and MEC are emerging endlessly and unable to meet the requirements of the complex battle field environment (e.g., low latency and bandwidth and high error rate and dynamic) that are crucial for future wars.

2.1. Mobile Cloud Computing. Mobile cloud computing uses the cloud computing technology on a mobile device, which brings the services like on-demand access and no on-premises software. MCC uses network capabilities alone to deliver the desired service to customers, which could permit to reserve network bandwidth confirming timely delivery of information to customers. The typical architecture of MCC is shown in Figure 1.

There are various researches about MCC proposed in the literatures focusing on the computation offloading and resource scheduling. Guo et al. [6] provided an energy-efficient dynamic offloading and resource scheduling (eDors) policy to reduce energy consumption and shorten application completion time. Chen et al. [7] proposed a game theoretic approach for the computation offloading decision-making problem among multiple mobile device users for mobile edge cloud computing (MECC). Jo et al. [8] proposed a hierarchical cloud computing architecture to enhance performance by adding a mobile dynamic cloud formed by powerful mobile devices to a traditional general static cloud, which increased the overall capacity of a mobile network through improved channel utilization and traffic offloading

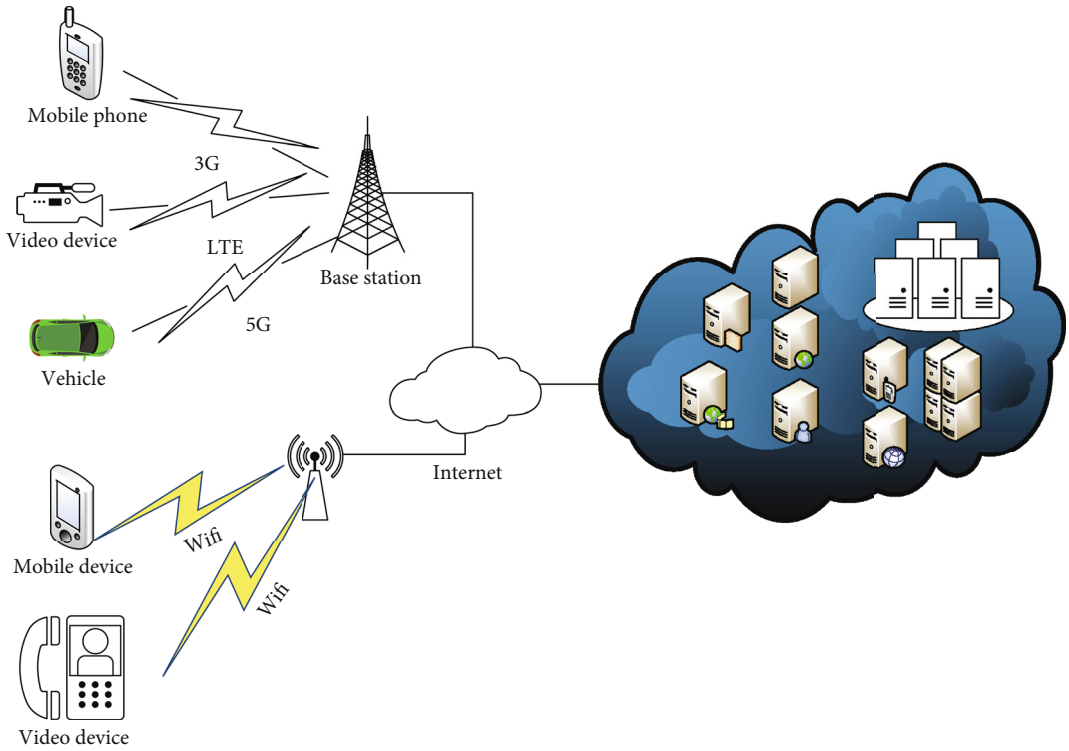


Figure 1: Typical architecture of mobile cloud computing.

from LTE-Advanced to device-to-device communication links. Han et al. [9] developed a unified framework that minimizes the overall outage probability in various mobile computation offloading scenarios. Miao et al. [10] put forward a new intelligent computation offloading based on MECC architecture in combination with artificial intelligence (AI) technology with the increasing requirements and services of mobile users, and an offloading strategy of simple edge computing is no longer applicable to MEC architecture. In 2019, the U.S. DOD officially announced that Microsoft will be responsible for building a cloud computing system for the U.S. military with a project cost of up to 10 billion dollars. In the next 10 years, Microsoft will build a core cloud computing system to achieve more efficient communication for the U.S. Army. The vision of MCC is an autonomous digital environment for different mobile devices to obtain their computation, storage, services, and other resources autonomously and efficiently anytime and anywhere [11].

2.2. Fog Computing. Fog computing is a concept proposed by Cisco Systems, introduced as a new network model to reduce data transfer within the IoT applications [12]. FC is a distributed paradigm that provides cloud-like services to the network edge, which leverages cloud and edge resources along with its own infrastructure [13]. FC involves the components of data processing or analysis applications running in distributed clouds and edge devices. It also facilitates the management and programming of computing, network, and storage services between the data center and terminal devices [14, 15]. In addition, it supports user mobility, heterogeneity of resources and interfaces, and distributed data analysis to meet the needs of widely distributed applications requiring low latency. Some architectures of the FC have been proposed, which were derived from the fundamental three-layer structure, extending cloud service to the network edge by introducing a fog layer between Internet of Things and cloud [16]. The typical architecture of FC is shown in Figure 2.

The hot topics in FC include a computation offloading and resource allocation scheme, scheduling policies, migration method, etc. Gao et al. [17] aimed to minimize the time-average power consumptions with stability guarantee for all queues in the system and exploited unique problem structures and proposed an efficient and distributed predictive offloading and resource allocation scheme for multi-tiered FC. Wu et al. [18] designed a value iteration algorithm of the semi-Markov decision process to maximize the total long-term reward for the task offloading problem of the vehicular fog and cloud computing system. Zeng et al. [19] considered a FC framework to support a software-defined embedded system, where task images lay in the storage server while computations can be conducted on either an embedded device or a computation server, which is significant to design an efficient task scheduling and resource management strategy with minimized task completion time for promoting the user experience. Bittencourt et al. [20] analysed the scheduling problem of FC, focusing on how user mobility can influence the application performance and how three different scheduling policies, namely, concurrent, FCFS, and delay priority, can be used to improve execution based on application characteristics. Osanaiye et al. [21] described an FC architecture, reviewed its different services

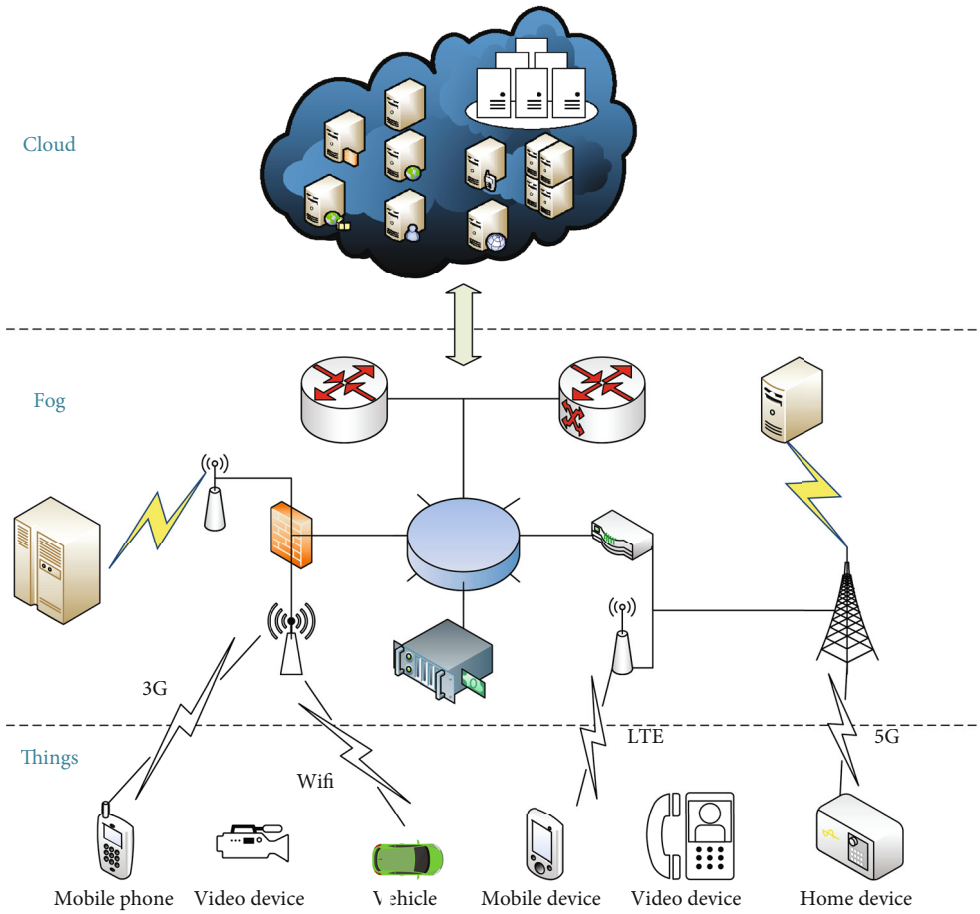


Figure 2: Typical fog computing network architecture.

and applications, and presented a conceptual smart precopy live migration approach for VM migration to estimate the downtime after each iteration to determine whether to proceed to the stop-and-copy stage during a system failure or an attack on a FC node.

FC brings many advantages including enhanced performance, better efficiency, network bandwidth savings, improved security, and resiliency. FC is not a replacement for cloud computing but extends the computation, communication, and storage facilities from the cloud to the edge of the networks [22], which is foreseen as a next computing paradigm and can be applied to a wide range of network applications.

2.3. Mobile Edge Computing. Mobile edge computing is a promising paradigm to offer the required computation and storage resources with minimal delays because of "being near" to the users or terminal devices. MEC is aimed at bringing cloud resources and services at the edge of the network, as a middle layer between the terminal user and cloud data centers, to offer prompt service response with minimal delay [23]. MEC refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services, which defines "edge" as any computing and network resources along the path between data sources and cloud data center. For example, a smartphone is the edge between body things and cloud, a gateway in a smart home is the edge between home things and cloud, and a microdata center and a cloudlet are the edge between a mobile device and cloud [24]. Many experiments are also depending on the environment of edge computing. The typical architecture of mobile edge computing is shown in Figure 3.

In recent years, the researches focus on edge computing including resource management, offloading strategy, and QoS enhancement. Mao et al. [25] provided a comprehensive survey of the state-of-the-art mobile edge computing (MEC) research with a focus on joint radio-and-computational resource management. Mao et al. [26] investigated a green MEC system with energy harvesting devices, developed an effective computation offloading strategy, and proposed a low-complexity online algorithm. Mach and Becvar [27] surveyed the existing concepts integrating MEC functionalities to the mobile networks and discussed current advancement in standardization of the MEC. Taleb et al. [28] proposed an approach to enhance users' experience of video streaming in the context of smart cities, whose approach relies on the concept of MEC as a key factor in enhancing QoS. Xu et al. [29] developed a collaborative method, named CQP, for the quantification and placement of edge servers (ESs). Zhou et al. [30] proposed a latency-aware microservice mashup

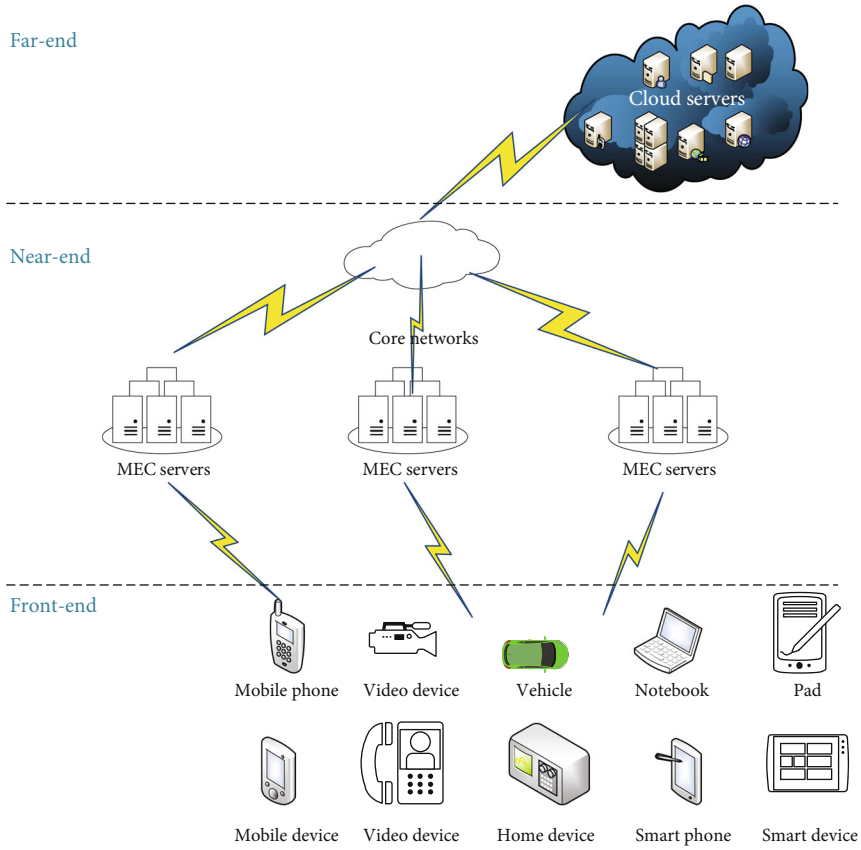


Figure 3: Typical architecture of mobile edge computing.

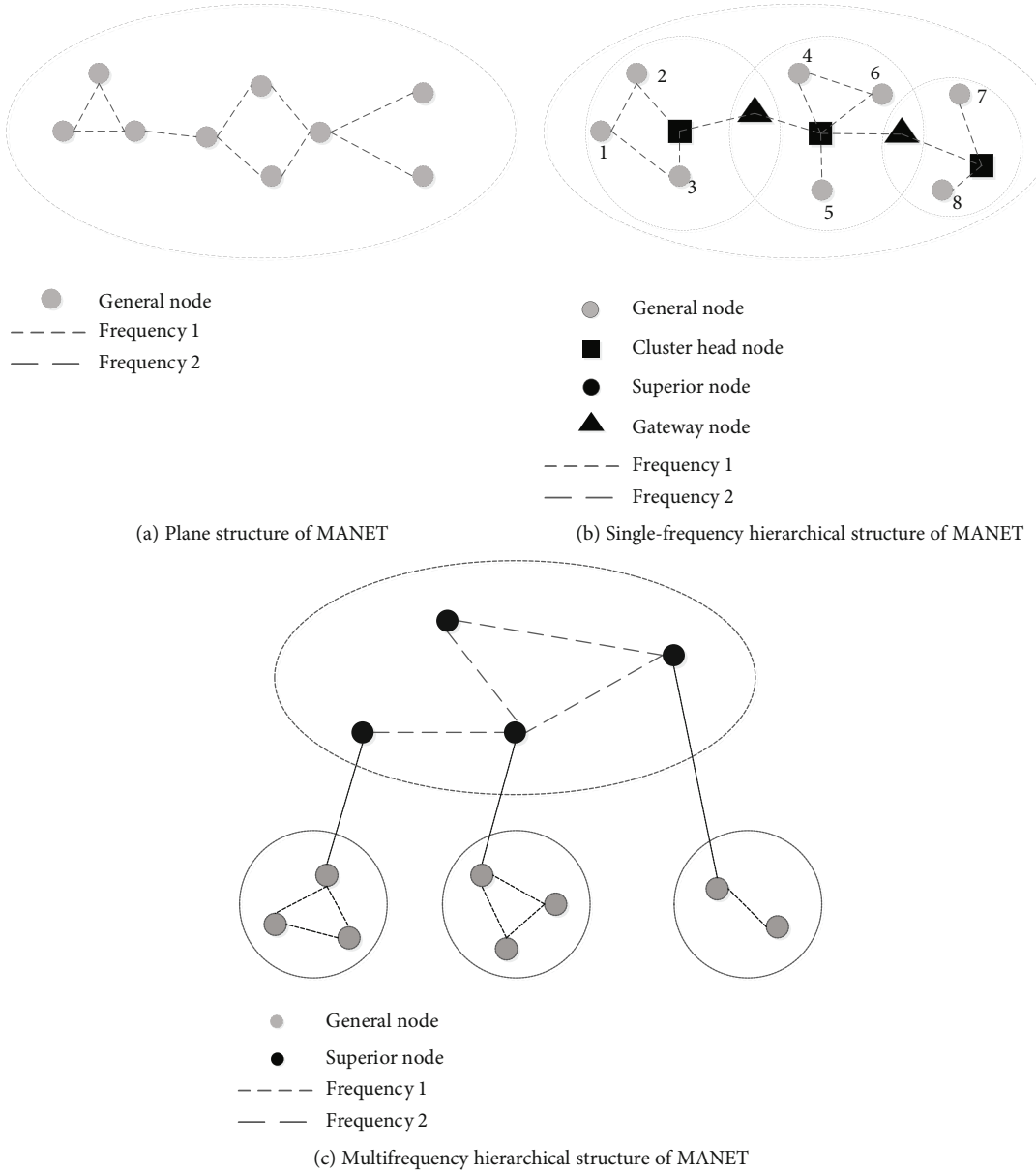
approach and the NP-hardness of the problem by reducing it into the delay-constrained least cost problem. Cao et al. [31] proposed a MEC-based system, in line with a big data-driven planning strategy, for charging stations.

MEC wants to put the computing at the proximity of data sources, has several benefits compared to the traditional cloud-based computing paradigm [32], and allows edge nodes to respond to service demands, reducing bandwidth consumption and network latency.

2.4. Mobile Ad Hoc Network. A mobile ad hoc network consists of multiple communication devices and terminals with transceivers, which can complete the communication process without infrastructure [33–36]. A multihop autonomous system built by MANET nodes has the advantages of no center, self-organization, and rapid networking [37]. The distributed network control and scalability of MANET bring great flexibility and convenience to the network deployment and practical applications [38]. However, it is difficult to predict the topology changes of MANET with the low frequency efficiency and large transmission delay [39]. The three typical architectures of MANET are shown in Figure 4.

The MANET researches focus on the routing protocol, QoS, energy management, etc. Zhang et al. [40] proposed a new greedy forwarding improvement routing method for MANET to calculate the reliable communication area and evaluate the quality of the link according to the relative displacement between the nodes and the maintenance time of the link. Sharifi and Babamir [41] presented a new clustering method and considered the good performance of Evolutionary Algorithms (EAs) good in finding proper head clusters and a specific EA-based method named Imperialist Competitive Algorithm (ICA) via numerical coding by considering the high efficiency of clustering methods among the routing algorithms. Zhou et al. [42] proposed a traffic-predictive QoS on-demand routing (TPQOR) protocol to support QoS bandwidth, delay requirements, channel assignment, and reuse schemes to reduce the channel interference and enhance the channel reuse rate. Ubarhande et al. [43] proposed a distributed delegation-based scheme to identify and allow the only trusted nodes to become part of the active path to improve the packet delivery ratio, packet loss rate, throughput, and routing overhead. Khosravi et al. [44] focused on the underwater communication and proposed a new method, using an intelligent 3D random node removal mechanism considering the traffic status of the network, to improve energy efficiency of the underwater acoustic wireless sensor networks and network reliability and better network lifetime. Pal and Jolfaei [45] proposed a software-defined wireless sensor network (WSN) architecture which conserves energy by applying asynchronous duty cycling.

MANET is conceived for military applications and aimed at improving battlefield communications and survivability originally; multihop MANET has lately been proposed in many civil scenarios, which is still difficult to achieve the large-scale applications [46]. As far as the application



(a) Plane structure of MANET

(b) Single-frequency hierarchical structure of MANET

(c) Multifrequency hierarchical structure of MANET

Figure 4: Typical architecture of MANET.

environments of these networks increase, the traditional communication paradigms of MANET need adequacy [47].

2.5. Delay/Disruption-Tolerant Network. A delay/disruption-tolerant network is proposed by DARPA and Internet Research Task Force (IRTF) to solve the communication problems in the network such as frequent interruption of communication links and large transmission delay [48–51]. DTN can be widely used in the area of interplanetary networks, military communications, wildlife tracking system, village-to-village networks in developing regions, vehicular networks, and nomadic communication networks [52]. In these networks, because of the frequent movement of network nodes, communication attenuation, and interference, the limitation of energy and storage space of nodes results in the long-time interruption of the network link, large delay, and high packet loss rate, which leads to the failure of the normal communication process and poor transmission performance [53, 54].

The research activities in DTNs recently are being investigated for the design of an application/convergence layer, routings, congestion/flow control, and security strategies, which are briefly introduced as follows: Banerjee et al. [55] presented a hardware and software architecture for energy-efficient throw boxes in DTNs and a novel paradigm for power management in DTNs provided for more efficient neighbor discovering by detecting the mobility of other nodes at a minimum cost and predicting the cost and opportunity of each possible contact, which can intelligently choose the most fruitful contact opportunities and limit the number of opportunities to meet energy constraints. Seligman et al. [56] proposed a congestion management solution of the last

Table 1: Summary of MCC, FC, MEC, MANET, and DTN.

Features	MCC	FC	MEC	MANET	DTN
Target user	Mobile user	Mobile user	Mobile device	Mobile device	General device
Server number	Few	Large	Large	Large	Large
Server deployment	Centralized	Near edge	Network edge	Device edge	Device edge
Reliability	High	Low	Low	Low	Low
Location awareness	Yes	Yes	Yes	Yes	Yes
Real-time interaction	Support	Support	Support	Support	Support
Latency	Low	Low	Low	High	High
Jitter	Low	Low	Low	High	High
Power	Ample	Limited	Limited	Limited	Limited
Storage capacity	Ample	Limited	Limited	Limited	Limited
Offloading	Yes	Yes	Yes	No	No
Programmable	No	Yes	YES	No	No
VM support	Yes	Yes	Yes	No	No

form for storage routing by employing nearby nodes with available storage to store data that would otherwise be lost when given uncontrollable data sources, which determines a collection of messages and neighbors to which they migrate using a set of locally scoped distributed algorithms, possibly incorporating loops that are known to be optimal for some DTN routing scenarios, and decouples storage management from global DTN route selection. Whitbeck and Conan [57] proposed a hybrid DTN-MANET routing protocol using DTN between disjoint groups of nodes while using MANET routing within these groups, which is fully decentralized and only makes use of topological information exchanges between the nodes. Mao et al. [58] proposed a new routing protocol, called scheduling-probabilistic routing protocol, using history of encounters and transitivity, and calculated the delivery predictability according to the encountering frequency among nodes to improve performance in both storage and transmission in DTN. Table 1 summarizes the features associated with MCC, FC, MEC, MANET, and DTN.

It can be seen from Table 1 that the current MCC, FC, MEC, MANET, and DTN have their own characteristics and advantages, but these computing paradigms are difficult to adapt to the requirements of high dynamic, low delay, and fast response at the tactical edge in the future.

3. The Features and Visions of DCOMP

The highly dynamic and complex battlefield environment results in the weak connection and highly dynamic characteristics of the tactical network, which puts forward strict requirements for the underlying computing and network infrastructure of the network. In the current technology, users usually rely on a large and highly shared data center to send data (such as image, video, or situation information) back to the data center for processing tasks with a large amount of computation. However, the rapid changes in the battlefield environment, the cost, and the delay of such backhaul may be problematic, especially when the network throughput is severely limited or user applications need near real-time response. Using the ability of "locally" available computing resources (from the perspective of latency, available throughput, task related, etc.) to calculate replication tasks could significantly improve the performance of applications and reduce the processing risk of tasks.

DCOMP provides a new way to improve the computing and communication capabilities in the harsh battlefield environment, which seeks scalable and robust communication and computing systems to meet users with competitive needs to safely and execute computing tasks collaboratively on a large number of heterogeneous computing devices running on a highly variable and degraded network environment.

DCOMP is different from the traditional network architecture completely, which does not regard the devices as the nodes of data transmission, but as distributed computing resources on the network. It can change the real-time dynamic network address at any time according to the needs. The distributed computing scenario consists of a group of NCPs with different computation abilities, such as a wireless access point, network router, handheld terminal, camera, and mobile computer. [59], as shown in Figure 5.

The concept of DCOMP has only been proposed in recent years, and systematic research on DCOMP is still relatively lacking, especially for network programmable protocols, computation offloading, task decomposition, etc.

García-Valls et al. [60] identified the new computing paradigm called social dispersed computing, analysed the key themes, and gave the outlook on its relation to agent-based applications, and examples include next-generation electrical energy distribution networks, next-generation mobility services for transportation, and applications for distributed analysis and identification of nonrecurring traffic congestion in cities. Hu and Krishnamachari [61] proposed a throughput optimized task scheduler, targeting applications (such as computer vision and video processing) where input data are continuously and steadily fed into the execution pipeline for DCOMP to perform computation on the edge leading to significant reduction in communication with the remote cloud. Fujikawa et al. [62] described DCOMP as a new vision of joint computation and communication resource

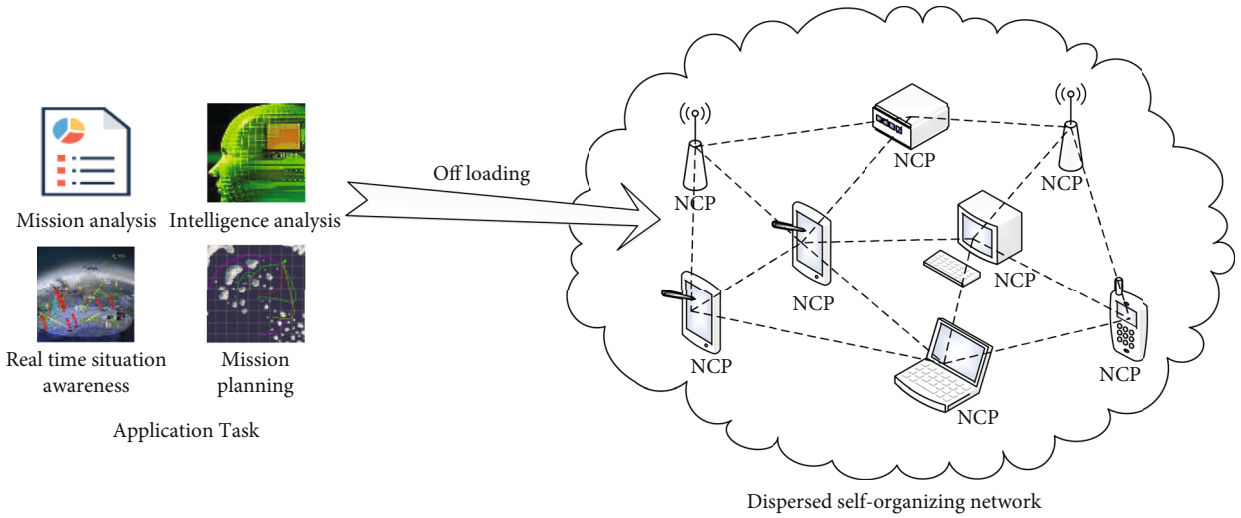


Figure 5: Schematic diagram of DCOMP.

management that goes beyond the end-to-end and client-server model of the current Internet and a new resource-centric architecture that leverages the diversity of networked computation points within the network and the heterogeneity of network links and protocol stacks that connect them. Yang et al. [63] considered the problem of task scheduling for such networks, in a dynamic setting in which arriving computation jobs are modelled as chains with nodes representing tasks and edges representing precedence constraints among tasks, and proposed a model motivated by significant communication costs in dispersed computing environments, and the communication times are taken into account. Knezevic et al. [64] designed a runtime scheduling software tool for DCOMP, which can deploy pipelined computations described in the form of a Directed Acyclic Graph (DAG) on multiple geographically dispersed compute nodes at the edge and in the cloud. Nguyen et al. [65] studied file transfer times between geographically dispersed cloud computing points using SCP (secure copy) and demonstrated via a set of real-world measurement experiments that the end-to-end file transfer time in a dispersed computing environment can be modelled as a quadratic function of the file size. Ghosh et al. [66] presented a container orchestration architecture for DCOMP and its implementation in an open-source software called Jupiter, which automates the distribution of computational tasks for complex computational applications described as a DAG to efficiently distribute the tasks among a set of networked compute nodes and orchestrates the execution of the DAG thereafter.

We believe that DCOMP has the following characteristics and advantages [67–69]:

(i) DCOMP should have the ability to coordinate various computing resources in heterogeneous networks for collaborative computing according to specific tasks and environment

(ii) DCOMP nodes have programmable capabilities and can respond dynamically according to the change of the network conditions, and the overhead associated with probing or message transmission between nodes must not significantly reduce throughput to support scalable, robust operation

(iii) DCOMP can quickly sense network bandwidth and topology changes with the movement of the nodes to ensure fast response to data service requests without having to send the data back to the rear data center to process the data locally, which can reduce the delay in data processing and improve the real-time capability of the combat system

(iv) DCOMP has the ability of crossing heterogeneous computing platforms to achieve more computing capabilities, by performing centralized task distribution. The various network components, radios, smartphones, sensors, and portable cloud computing equipment in DCOMP can be reasonably applied in the programmable execution environment to maximize the computing capacity of the battlefront

4. Network Model, Channel Allocation, and Forwarding Control Mechanism of DCOMP

In the future, the hostile battlefield needs to establish a centerless, self-organized tactical communication network consisting of tactical radio stations and individual handheld/-vehicle/airborne/shipborne wireless terminals, which can be regarded as a MANET.

The network of DCOMP is a centerless, multihop, self-organizing, infrastructure-free, peer-to-peer communication network composed of various wireless communication nodes without a strict central node. All nodes use on-demand routing protocols and a proactive routing mechanism to coordinate their behaviours, which form an independent network quickly and automatically and perform calculations

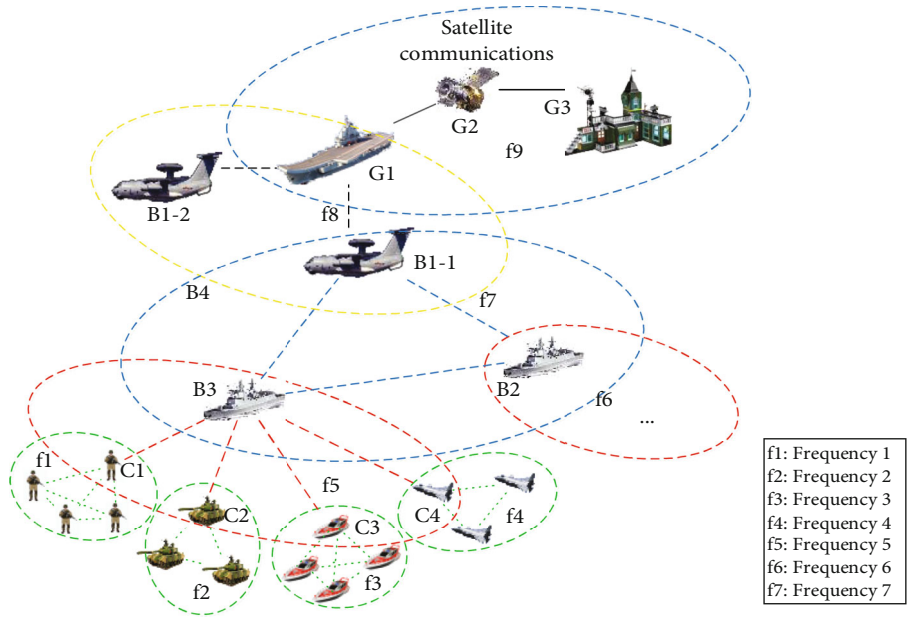


Figure 6: The typical structs of DANET.

spontaneously and collaboratively. We call them the dispersed ad hoc network (DANET).

4.1. Network Model of DANET. Each tactical end node in the DANET has both the role of a router and computing host. The nodes in a DANET environment need to run combat-oriented applications and corresponding routing protocols as routers. The DANET is different form MANET for the reasons of topology changing faster, moving at different speed, lower latency required, and nodes joining or exiting the net at any time.

It is difficult to communicate the real-time battlefield environment, situation, and combat command data through DANET in a hostile battlefield combat environment. The typical structs of DANET in the battlefield environment is shown in Figure 6, which is a hierarchical network composed of independent communication subnets (in each virtual box) with the original characteristics maintained. A node is selected as a cluster head in each subnet, and the cluster head is not only a member of the subnet at this level but also a member of a superior network.

This struct of DANET enables battlefield information to be sent directly from the lowest level to the upper level and enables command orders to be sent directly from the upper level to the lowest level, reducing manual forwarding though intermediate links and improving timeliness. The multifrequency hierarchical structure adopted by the DANET implements functions such as command communication, situational forwarding, and backbone network connection.

4.2. Channel Sharing and Allocation of DANET. The network is shown in Figure 6, and each subnet uses the same frequency and shares the same channel. Nodes can perform unicast, multicast, and broadcast communications in the same subnet. Different subnets use different frequencies and channels to avoid interference between the nodes and improve the frequency reuse rate. The cluster head (such as the nodes of C1, C2, C3, C4, B3, B2, B1-1, and G1 in Figure 6) is responsible for forwarding data to achieve communication between different subnet nodes. The cluster head has a duplex frequency and can work on two channels at the same time, of which one is responsible for communicating with the subnet and the other channel joins the upper-level network. Therefore, the cluster head plays a role as the gateway node connected to other subnets.

The DANET can use fixed channel allocation to solve the problem of channel allocation and enable the network to achieve a good balance in reducing interference and improving network connectivity. The channels of the entire network are uniformly allocated before communication, and the available channels are divided into working channels and standby channels allocated to each subnet. The different adjacent subnets have different channels to avoid interference. If the working channel is subject to external interference, each node within the subnet will use the standby channel according to the relevant agreement. At the same time, multiple subchannels are allocated to each subnet, and the channel is changed when external interference is encountered or at a specified time according to the agreement. Each subnet is used strictly in accordance with the rules of the agreement to avoid problems of blind channel use and channel resource contention.

In the tactical edge network, a large number of real-time applications (such as voice, video, or urgent tasks) are generated with the constant change of network requirements. In the traditional MANET, the mobile nodes move in the network at will, which will cause strong dynamic characteristics. The topology of DANET work changes with the movement of nodes in unpredictable ways and at unpredictable speeds, which may cause communication interference between adjacent nodes leading to seriously affecting throughput and transmission delay of the network.

Table 2: Comparison of different routing protocols.

Protocol	AODV	OLSR	DSDV	CGSR	LANMAR
Proactive/reactive routing	Reactive	Mix	Proactive	Proactive	Proactive
Location based	No	Yes	No	No	Yes
Hierarchy	Plane	Hierarchy	Plane	Plane	Hierarchy
Multicast	Yes	Yes	No	No	No
Routing discovery delay	High	Low	Low	Low	High
Routing overhead	Low	General	High	High	Low
Advantages	Compression control information	Available immediately	Respond quickly	Small time delay	Small routing table size
Disadvantages	Best performance at low congestion & high density	More resource than AODV	More useless routing information	Weak adaptability of topological change	Large storage control

The traditional IEEE 802.11 standard uses the shared channel model, and the interference increases with the increase in the number of mobile nodes, resulting in a significant decrease in network performance. In the environment with dense communication nodes, the application of Multiple Access Control Protocol (MACP) will adversely affect network performance. When a mobile node enters the communication range of another pair of node pairs, each node can carry out channel switching, which will cause the mobile network to continuously receive interference, and channel allocation is needed for dynamic channel management. In order to effectively solve the channel interference problem caused by exposed nodes in DANET, the channel allocation control algorithm with power control can be used to dynamically negotiate the channel to carry out adaptive channel allocation. It can realize multiple communications of different channels in the same area, reduce the negative effect caused by the exposed node problem, and improve the capacity of the network [70, 71].

4.3. Routing Protocol and Forwarding Control of DANET. Due to the feature of poor connection and high dynamic of the DANET, the routing protocols of traditional MANET cannot adapt to the low latency and high reliability requirements at the tactical edge. The routing protocols of DANET must satisfy the actual operational requirements of fast topology changes, high real-time performance, and strong anti-interference capabilities. Therefore, it is necessary to establish a dispersed organization-aware network programmable routing protocol and forwarding control strategy in a harsh battlefield environment.

A hierarchical routing protocol needs a complex cluster head election algorithm, and the cluster heads are easy to become the bottleneck of the network for the reason of most of data forwarded by cluster heads. The typical hierarchical routing protocols include CGSR (Cluster-head Gateway Switch Routing Protocol), HSR (Hierarchical State Routing Protocol) [62], LANMAR (Landmark Ad Hoc Routing Protocol) [72], etc., and the common table-driven routing protocols are DSDV (Destination-Sequenced Distance Vector) [73], WRP (Wireless Routing Protocol) [74], OLSR (Optimized Link State Routing) [75], AODV (Ad hoc On-demand Distance Vector Routing) [76], etc. It can be found that there are multiple routing protocols used in the current MANET, and different types of protocols have different mechanisms with their own characteristics and different environmental adaptabilities. Table 2 shows the different application scenarios and advantages and disadvantages of the typical routing protocols.

Form Table 2, we can see that the different types of routing protocols have different mechanisms in MANET; each protocol has its own characteristics and adapts to different environments. For example, WRP and GSR have fast convergence speed, but the node burden is heavy, which is not suitable for the network with strict requirement on node energy; OLSR is suitable for the networks with small mobility, frequent internode communication, and large scale; AODV has better performance in high-load, high-mobility networks. The AODV and OSLR are typical and mature routing protocols of the MANET, which are very suitable for the traditional MANET. Because the topology change speed in traditional MANET is less than the tactical edge in future wars, the original AODV and OLSR routing protocols are no longer suitable for this highly dynamic tactical edge environment. We propose a routing protocol combination of OLSR and AODV to solve hierarchical routing of networks in harsh battlefield environments for DANET. The routing protocol of AODV can be used as a routing protocol for small wireless ad hoc networks within a subnet, and OLSR as a routing protocol between subnets. The original AODV and OLSR routing algorithm is difficult to adapt to complex battlefield environments and needs to introduce an improved algorithm of OLSR and OLSR protocols for DANET.

4.3.1. Improvement of OLSR Protocol Based on Dynamic Link Cost. Applying the OLSR protocol as a tactical edge backbone network (between tactical edge subnets) is more conducive to the rapid forwarding of command and control information. The workflow of the OLSR protocol is shown in Figure 7.

However, in the MPR (Multipoint Relays), selection of the original OLSR protocol only considers the state of physical connectivity of the link, but the bandwidth status of the link is not considered. In the path calculation, the original

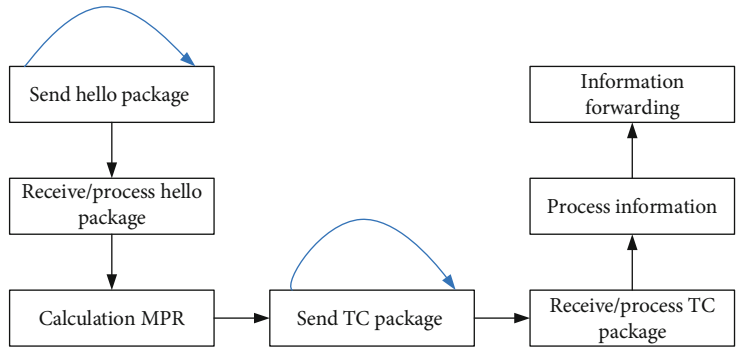


Figure 7: Workflow of the OLSR protocol.

OLSR protocol only considers the minimum number of hops without link latency, but the delay of the link path is not considered. In the harsh battlefield environment, the amount of various types of burst data is very large transmitted along the shortest path, which can easily cause network congestion, resulting in the decrement of network throughput and packet transmission rate and increment of transmission delay. The original OLSR routing protocol only takes the physical connectivity of the link into account, not considering the bandwidth of the link, which obviously cannot meet the requirement of high service quality in DANET. Therefore, in order to improve this situation of possible congestion, the original OLSR protocol needs to improve by considering link cost of message flooding and path selection to avoid the bottleneck of the network for data transmission.

In order to implement the improved OLSR protocol based on the dynamic link cost, it is necessary to modify the Hello packet, TC packet, routing topology table, etc. and add "Cost" information into the package. In the complex battlefield environment, the delay requirement of information transmission is very strict. The dynamic link cost function should not be too complicated, which needs to adjust according to changes of network conditions (such as bandwidth, delay, packet loss rate, and energy consumption) to reduce routing congestion. Therefore, the "Cost" can be set as

$$\text{Cost} = 1/\text{B\_link}, \tag{1}$$

where "Cost" is the link weight and "B_link" is the link bandwidth.

We can add "Cost" to the Hello packet (a), TC packet (b), and routing topology table (c), as shown in Figure 8.

In the Hello packet, the "Cost" is the cost of the link from the node to the neighbor node. In the TC packet, the "Cost" is the link cost between the node and the MPR node. In the routing topology table, the "Cost" is the link cost between "T_dest_Address" and "T_last_Address." During path calculation, the cost of the link is obtained from the network topology table, which is used as the link weight value, and the Dijkstra algorithm is called to obtain the optimized path after considering the link bandwidth status.

4.3.2. Improvement of the AODV Protocol Based on Dynamic Link Cost. Each router (the cluster head) in the DANET can access the node in the subnet, and the subnet can form multiple relatively small wireless ad hoc networks through wireless connections. The AODV protocol is used at the inside subnet of the MANET. The original AODV protocol has well performance when the nodes move faster and a large amount of business data uses the minimum hop path between the source node and the destination node for routing selection. The minimum hop path is not necessarily the best path, which can cause network congestion and affect network performance. Therefore, the routing algorithm based on dynamic link cost is also needed to improve the original protocol of AODV in DANET.

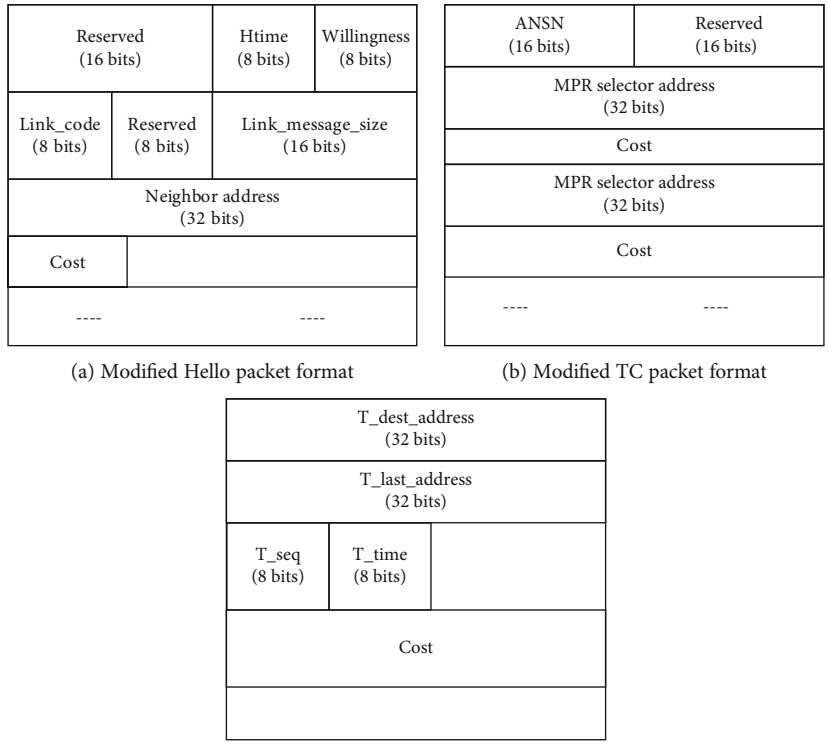
Define $\text{Path}(s, d)$ as all feasible paths from source node s to destination node d, set k as the number of elements in the path, and $\text{Path}_i(s, d)$ represents a feasible path from node s to node d $(i \le k)$. The hop count of the path is $\text{hop}(\text{Path}(s, d))$, and e_{ij} represents the link between nodes V_i and V_j. Set the bandwidth of the link to be B_link, the real-time bandwidth of the link is B_use, and the request bandwidth of the service is B_req.

The improved model must satisfy

$$\begin{aligned} \text{Bf} &= \min\{B\_\text{req} - (B\_\text{link} - B\_\text{use})\}, \\ \text{Br} &= \max\left\{\frac{B\_\text{link} - B\_\text{use}}{B\_\text{link}}\right\}. \end{aligned} \tag{2}$$

(i) Bf: bandwidth fragment. When selecting a link, one should try to select a link with a request bandwidth that is close to the actual available bandwidth value, so that the bandwidth fragmentation is as small as possible to achieve the purpose of improving bandwidth utilization

(ii) Br: bandwidth ratio. Under the condition that the requested bandwidth and the actual available bandwidth are as close as possible, the links with a large ratio of the actual available bandwidth to the link's own bandwidth should be selected to reduce the probability of congestion on the link caused by sudden bandwidth requirements



(a) Modified Hello packet format

(b) Modified TC packet format

(c) Modified routing topology table

Figure 8: Add dynamic link "Cost" to improve the OLSR protocol.

Based on the above two indicators, the link cost function is determined as

$$\text{Cost} = \begin{cases} \dfrac{\alpha\left|(B_{\text{link}} - B_{\text{use}})/B_{\text{link}}\right| + \beta \lg \left|B_{\text{reg}} - (B_{\text{link}} - B_{\text{use}})\right|}{\text{Hop}(\text{Path}_i(s,d))} (B_{\text{r}} \neq 1), \\ \dfrac{1}{\beta\left|B_{\text{reg}} - (B_{\text{link}} - B_{\text{use}})\right|} (B_{\text{r}} = 1), \end{cases} \tag{3}$$

where α and β are adjustable proportion coefficients of bandwidth fragmentation and bandwidth ratio, and the size of the two can be adjusted according to the demand; $\text{hop}(\text{Path}_i(s,d))$ is the number of hops from node s to node d. The link "Cost" value can be adjusted according to the link bandwidth status and traffic demand status in real time.

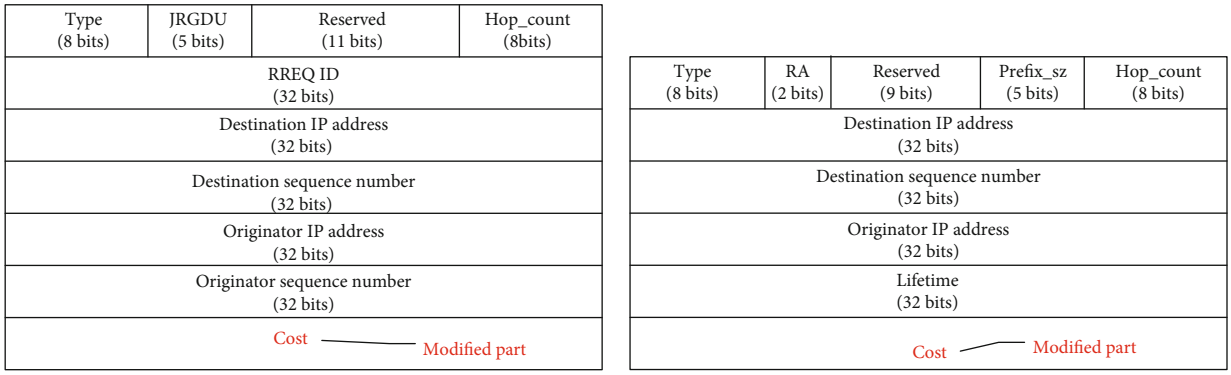
The "Cost" is added to the routing request packet and response packet, respectively, and the cost information of the path can be obtained when the routing calculation is finally performed, as shown in Figure 9.

4.4. Congestion Control Strategy of DANET. The DANET also can be considered a typical DTN, and the congestion control of this kind of network has the following challenges: (1) there may be no communication opportunities in the future, (2) the received save transmission messages cannot be dropped, (3) reserve a cache space for different service types, and (4) reject new connections when storage is full. The research of congestion control strategies in DTN mainly includes node packet loss strategies and message weight calculation models. The entire process of the congestion control strategy includes the processes of receiving messages and comparing weights of sending messages. To judge the message discarding according to the current congestion and weight comparison, we design a specific method shown in Figure 10. If there is no congestion, the message is put into the buffer; if it is congested, the preservation weight of the message is compared with the preservation weight of the node. If the message weight is greater than the node weight, the message with the smallest weight in the node buffer is discarded until there is enough space in the buffer to accommodate the new message.

For the calculation of the weight of a message, the size of the message, the initial lifetime, and the remaining survival time are mainly considered. Formula (4) gives the calculation method of message weight:

$$W_{ri} = \frac{\text{TTL}_{mi}}{\text{TTL}_0} \times \frac{\text{BS}_j}{\text{S}_{mi}}, \tag{4}$$

where W_{ri} represents the weight when the message i arrives at node j, TTL_{mi} represents the remaining survival time for message i, TTL_0 represents the initial lifetime of all messages in the network, BS_j is the buffer size of node j, and S_{mi} is the size of message i. As can be seen from Equation (4), the shorter the remaining survival time, the lower the probability of the message reaching the destination node; the smaller the weight of the message, the larger the size of the message; and



(a) Routing request packet

(b) Routing response packet

Figure 9: Add dynamic link "Cost" to improve the AODV protocol.

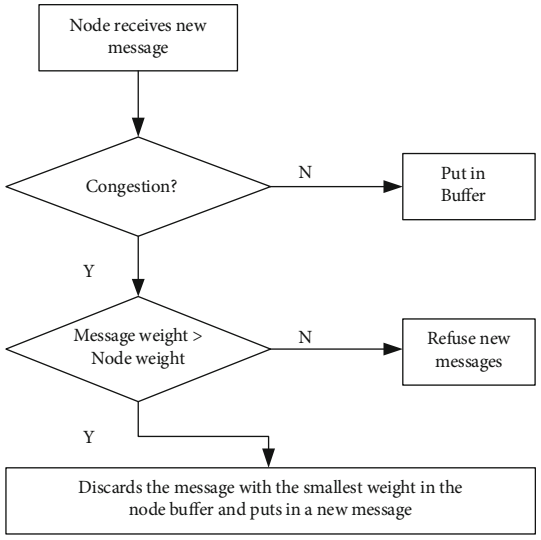


Figure 10: The process of the congestion control strategy for DANET.

the greater the preservation cost, the lower the transmission success rate and the larger the weight of the message.

The formula for calculating the weights of nodes is shown in

$$\mathrm{EW}_j = \frac{\sum_i^{M_j} W_{ri}}{M_j}, \tag{5}$$

where M_j is the total number of messages in the buffer of node j and EW_j is the average preservation weight of node j.

When a node tries to send a message, check whether there are any messages in the buffer that are smaller than the transmission time threshold. If it exists, it uses the greedy strategy to preferentially forward the message with the smallest transmission time; otherwise, it does not forward it. The transmission time threshold is derived from the node's moving speed and communication range.

For each node j, the transmission time threshold is difficult to calculate and can be estimated by

$$\mathrm{Flag}_j = \frac{S_{Lj}}{V_{Nj}} \times \delta, \tag{6}$$

where S_{Lj} represents the communication range of node j, V_{Nj} represents the moving speed of node j, and δ represents the weighted value of transmission. When the communication range of node j is larger, the communication time between other nodes and node j is longer. When the moving speed of node j is faster, the contact time between node j and other nodes is shorter.

Formula (7) gives the calculation method for the forwarding delay of message i:

$$W_{ij=\frac{S_{mi}}{V_{Tj}}}, \tag{7}$$

where W_{ij} represents the forwarding delay of the message i and buffered in node j, S_{mi} is the size of the message i, and V_{Tj} represents the data transmission speed of node j. The forwarding delay of message i is obtained by the ratio of the size of message i to the data transmission rate of node j. When the size of message i is larger, the transmission delay at a certain transmission speed is longer and the buffer of node j is larger, resulting in the greater cost of node j to forward message i and the lower success rate of forwarding. When the forwarding speed of node j is faster, the forwarding delay of each message in the buffer is shorter, resulting in the forwarding success rate being higher.

5. The Architecture Design of DCOMP

In the scenario of DCOMP, the communication bandwidth between different nodes is very limited and heterogeneous, and the computing nodes support different computing capabilities. Figure 11 shows the dataflow and architecture of DCOMP, which connects nodes with computing capabilities to the DANET to communicate directly or indirectly, greatly reducing the time for data transmission and processing. In the paradigm of DCOMP, the nodes are both consumers of data and producers of data.

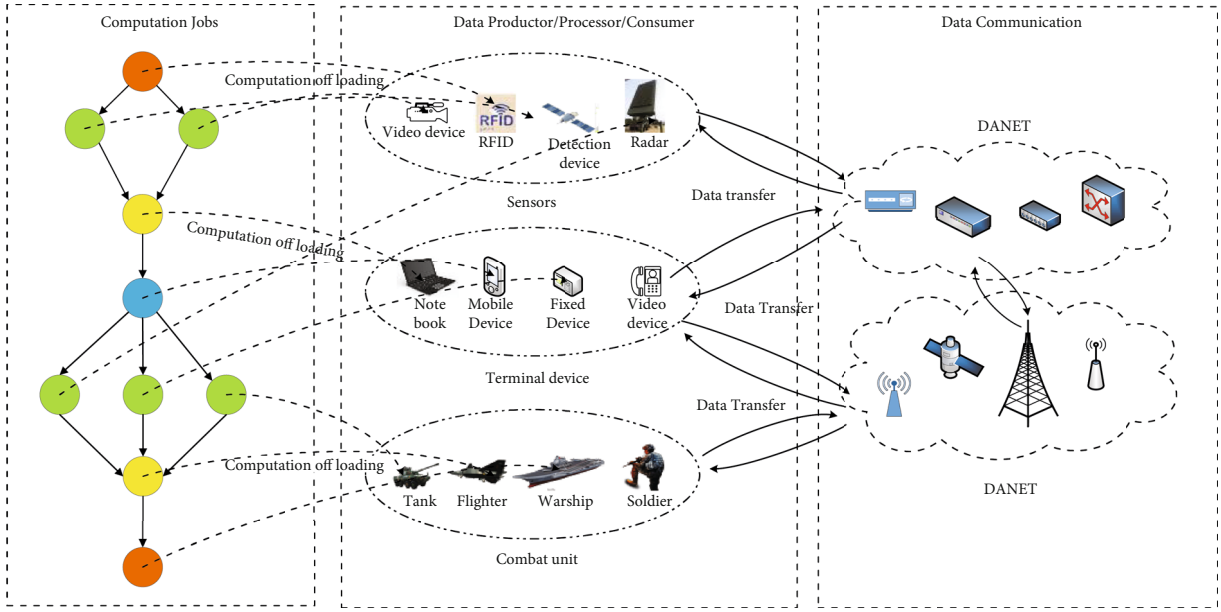


Figure 11: The dataflow and architecture of DCOMP.

As shown in Figure 11, the architecture of DCOMP can be seen that the data can be processed not only at local but also on geographical NCPs. These dispersed computing nodes can perform data collection/transfer/caching/processing, computation offloading, and task scheduling to realize the function of traditional distributed computing.

5.1. The Computing Core and Unit of DCOMP5. Aiming at the computing model in DANET, we intend to promote the computing model in the distributed environment, which expands the concept of the core of distributed computing and defines the role of NCPs as two types according to the available software and hardware:

(i) Computing core (CC): it mainly includes the main program running on the cluster head node, which is responsible for managing the execution process of the entire computing task and abstracted as the application computing core in DCOMP

(ii) Computing unit (CU): it mainly includes the real computing program running on general nodes, responsible for specific computing tasks and abstracted as a computing unit in DCOMP

Different from the traditional distributed computing architecture execution flow, the calculation of the main control program in CC and the calculation program in CU are performed at a different node. The CC may also assume the role of CU in DCOMP. Figure 12 shows the struct of the CC and CU.

When the main program (CC code) at the cluster head node (C1) starts the main program and executes the first calculation program function (xFuction1), C1 actively queries available computing resources (software and hardware) from the local resource database to schedule and allocate computing resources on a general node (Un) according to the request of the CC code. The main program notifies the management program on the general node (Un) of the code and data and starts the computing program (CU code) after offloading the code and data to Un and returns the status to the CC code after completing the calculation task to maintain the data consistency. The CC code continues to execute the second calculation program function (xFuction2). It still needs to apply for resources again and notify the assigned general node (Um) management program to offload the code and data and start the calculation. The CC code continues to execute the calculation process until the entire computing task is completed.

The CC code and the CU code are usually not on the same node in DCOMP, but their roles are interchangeable. The code in traditional distributed computing architecture is usually deployed by the master node to the slave node at one time. The master node controls the program fragments in the slave node to perform related calculations, and the calculation results are summarized in the master node regardless of the current position, resources, and status of the slave node. In the DCOMP, the CC is responsible for maintaining all flow and CU positions and resource information list and updated in time after the execution of the CC code to ensure that all calculation data are correctly accessed to maintain the data consistency.

5.2. The Software Stack of DCOMP. The software stack of DCOMP includes an application software layer, programming framework layer, resource management layer, network communication layer, operating system layer, and device layer, as shown in Figure 13.

The programmable computing framework includes the TCL (Tool Command Language) [77] interpreter and its programming model formed by the runtime environment. It provides a set of available interpretation instructions for programmers to write applications with distributed

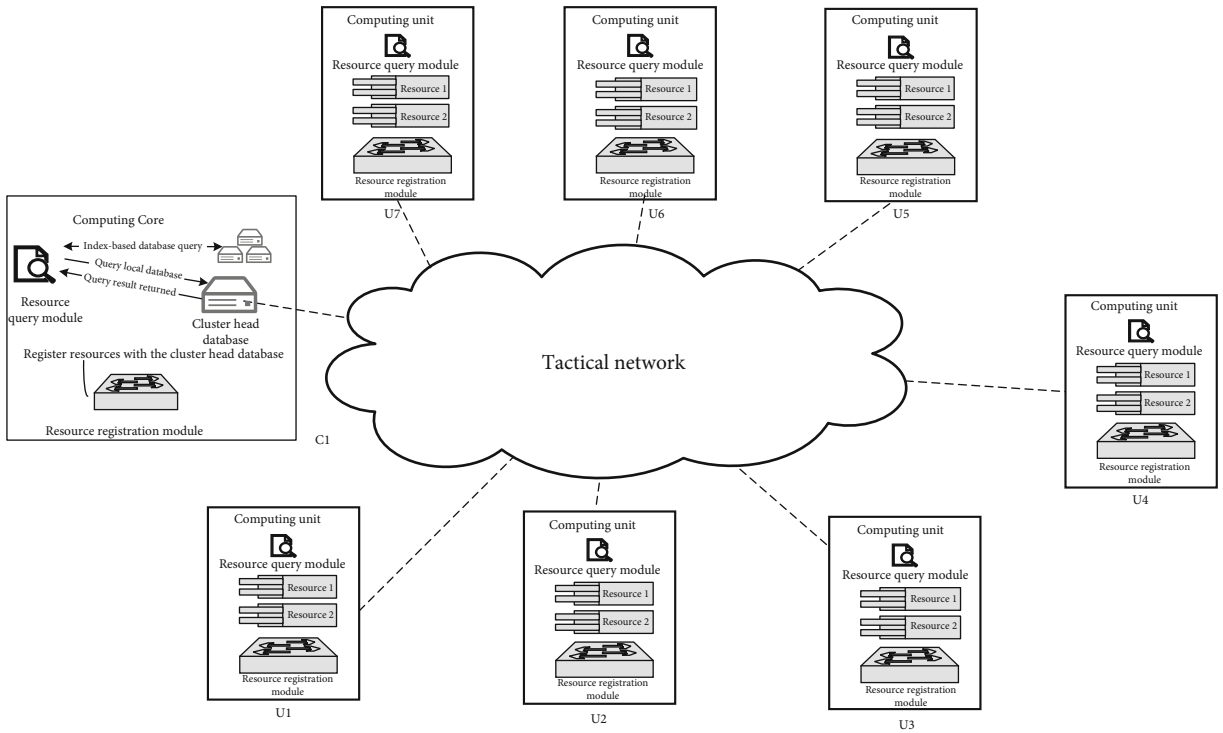


Figure 12: The struct of the computing core and computing unit.

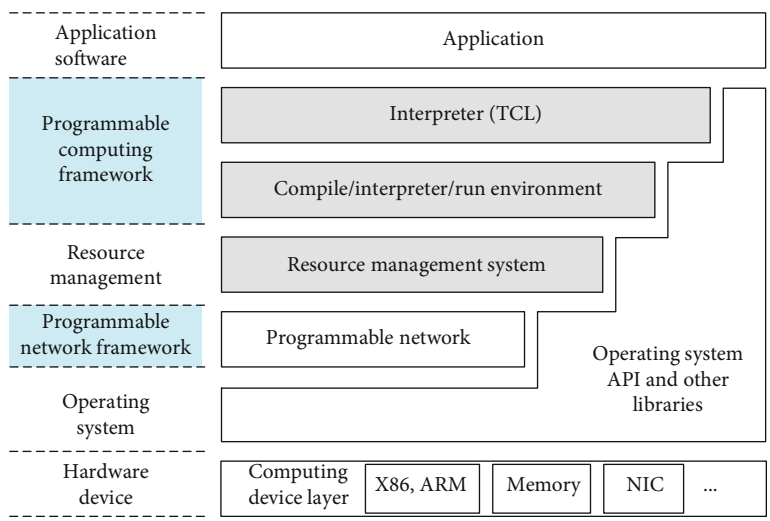


Figure 13: The software stack of DCOMP.

semantics. By using these interpretation instructions explicitly, users can concentrate on program performance optimization, such as the development of parallelism between CC and CU, without paying too much attention to details such as the heterogeneity of underlying resources, dynamic resource binding, and load balancing.

In the programmable network framework, network scheduling and forwarding control of NCPs in DANET is a difficult problem. Programmable network programming model and protocol specifications for interactive use between each node can provide network-aware programming interfaces and scheduling programming interfaces, connectivity and bandwidth utilization awareness, path state threshold calculation and failure analysis, real-time path changes, and other programmable capabilities [78, 79].

6. The Programming Model for DCOMP

Traditional distributed computing environments generally hard code all kinds of algorithms and software to each node and transmit commands with adjustable parameters to call and control the software of each node, which will not be able to meet the requirements of the rapid and flexible call of various resources in future war. Is it feasible to download the

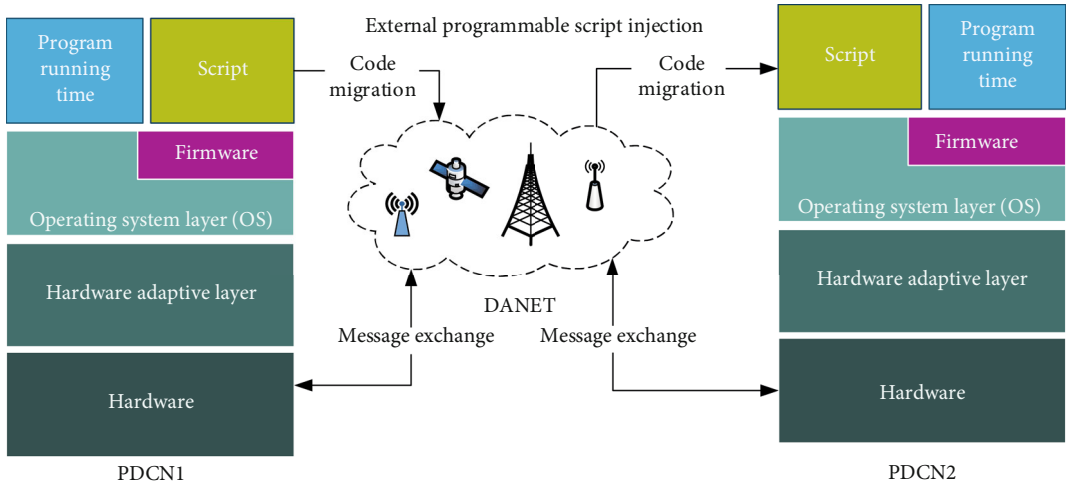


Figure 14: The framework of programmable of PDCN.

compiled executable image to each node? The answer is negative, because most nodes are sometimes inaccessible in physical resources or file transfer at a very high cost. Generally, the energy efficiency of transmitting executable compiled executable images to each node through the network is very low (high communication costs and limited node energy).

In the harsh battlefield environment, it is required that DCOMP be able to provide task computing as an aggregated whole, not just to provide services as a single node, needing to achieve DCOMP nodes that have the ability to be programmable dynamically. This means that an NCP connected to the network at any time will be able to inject instructions into the network to perform a given task. The instructions will call a single node based on the needs of the combat mission, network status, and physical resources.

6.1. Programmable Computing Model for DCOMP

6.1.1. Programmable Model. This paper proposes a programmable framework for DCOMP, which attempts to complete the computing tasks by a user-defined way in the dispersed networking environment. We called this kind of node a programmable dispersed computing node (PDCN). The framework of the programmable computing model is shown in Figure 14. The framework can be divided into the hardware layer, hardware adaptive layer (device driver), operating system (OS) layer, programmable software layer, and PDCNs in DCOMP to achieve data communication and computation offloading through DMANT.

(i) Hardware layer includes a wireless transceiver module, various sensors, timers, and other computing hardware resources

(ii) Hardware adaptive layer provides various types of equipment drives

(iii) Operating system layer provides all standard functions and services of the multithread environment required for the programmable software layer. The firmware provides the functions necessary for hardware resource calls, thread/file/communication operations, etc,

(iv) Programmable software layer provides the running time environment for the programmable language

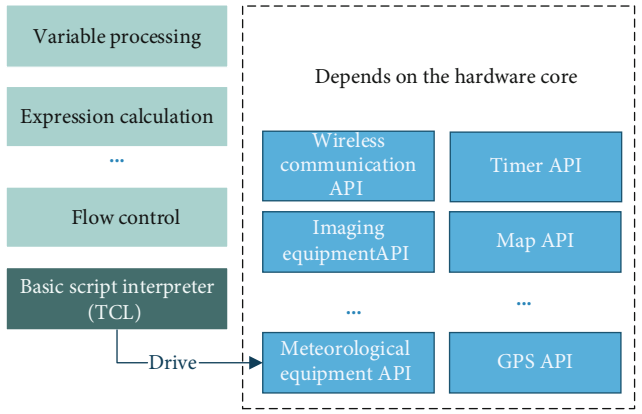


Figure 15: The composition of the programmable language.

6.1.2. Programmable Language. The basic idea of programmability is to make nodes have the ability to be programmable through control scripts, including the scripting language and the corresponding programming model. The scripting language needs to define and implement the appropriate functions/commands in order to use them as building blocks (such as the basic commands of a script). Each of these commands will abstract a specific task from the tactical end node (communication with other nodes, sensors to obtain battlefield data, etc.). The scripting language also needs to construct the syntax of the corresponding control script, including syntax for flow control (such as loops and conditional statements), syntax for variable processing, and syntax for expression calculation. Figure 15 shows the composition of a programmable language.

As we can see from Figure 15, the PDCN first parses the statement of the programmable code and then judges the

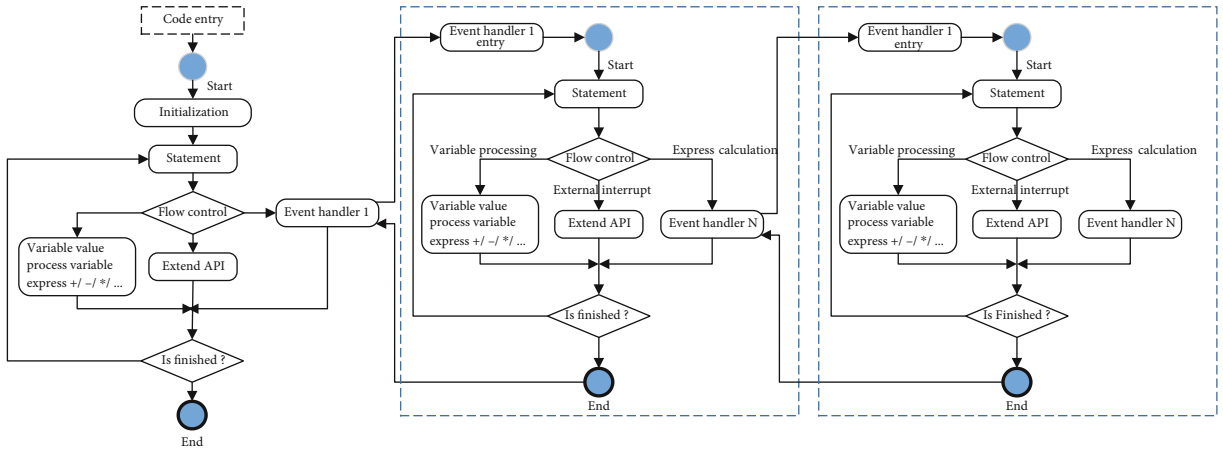


Figure 16: The workflow of the programmable model.

execution flow of the statement. If the statement is an "event handler," then the workflow will enter the function of the "event handler program." If the statement is "variable processing," "expression calculation," or "execute external API," the program will execute the corresponding operations, until the end of the program.

The basic script interpreter can use the open-source syntax interpreter TCL, which provides well extensibility and portability for programmable languages. All basic commands (such as switch, if, while, and other commands) can be defined as the new programmable script syntax using the standard TCL.

According to the model of DANET, the script can be viewed as a state machine affected by external events (like interrupt messages) including network messages from DANET, sensed data, and timer period timeout. The programming model is used to define an event progress, which determines the execution flow based on the current state, and the new event or state will be processed. Figure 16 shows the workflow of the programmable model of a typical PDCN.

6.1.3. Runtime Environment. As important as programmable languages and models is the runtime environment running the script. Different PDCNs provide different resources including different hardware and software resources.

For example, PDCN A has a transceiver and a magnetometer, PDCN B has two transceivers and a camera, and their operating systems are different. The programmable language framework solves the heterogeneity of PDCNs through abstraction and defines and adds arbitrary tasks in the runtime environment through the extended API provided by the abstract module/service interface constructed by the programmable framework.

All devices have fixed interfaces to operate on events through four commands: Query, Act, CreateEventID, and DisposeEventID. Query requires the device to obtain device hardware and software information; Act instructs the device to perform actions (modify some parameters of the device, perform some actions); CreateEventID describes the specific event that the device can generate, provides a name or ID for this event, and waits for the specific event returned with that

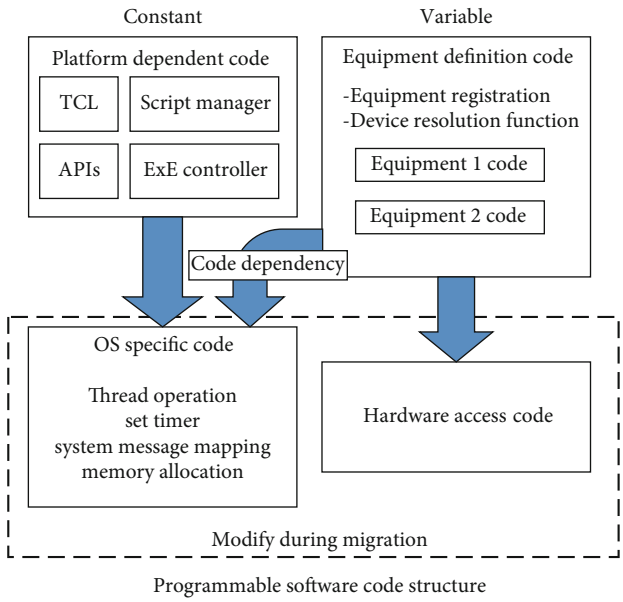


Figure 17: Programmable software code structure.

name in the command; and DisposeEventID unregisters the event and releases the resources.

Even two PDCNs have the same functionality, and the hardware or operating systems may be different. In order to facilitate the migration process, it is necessary to clearly separate the OS layer- and hardware layer-specific code from the solidified code and function definition code. In order to implement the functions defined in programmable software, it needs to identify the code's dependencies with the operating system and the hardware and create associated abstract constructor interfaces. The constructor interface is defined separately in the code file (such as .c file), and developers can easily identify and migrate a code to suit the MANET environment. The operating system needs to support for creating and starting threads/tasks, as well as for defining, publishing, and suspending message queues. In addition, the hardware access code is defined to realize low-level hardware resource call and manage. Figure 17 illustrates the code structure of the programmable software.

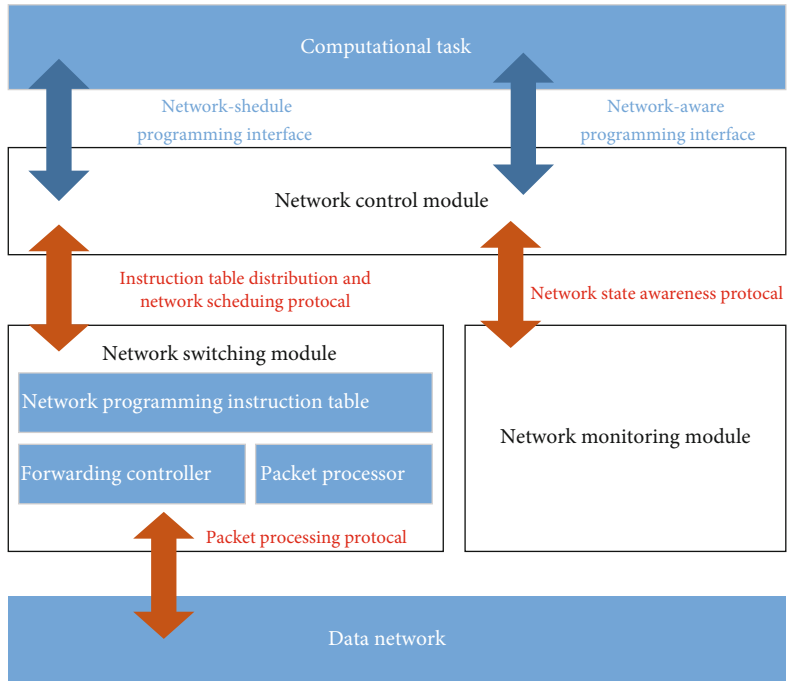


Figure 18: The components and dataflow of the programmable network model.

6.2. Programmable Network Model for DCOMP. In order to solve the network scheduling and forwarding control of PDCN in DANET, it is necessary to design a programmable network model for DCOMP and a protocol specification for interactive use among entities according to the model. The model provides the network-aware and scheduling programming interface, which supports programmability of connectivity and path bandwidth awareness, state threshold calculation and failure analysis, and real-time path choice. Figure 18 shows the components and dataflow of the programmable network model in DANET.

6.2.1. Network Control Module. The module provides network-aware and network scheduling programming entry and receives the task distribution plan issued by CC, which is combined with the network connectivity state diagram to generate the instruction table and then sent to the relevant forwarding controller and packet processor to implement network scheduling.

6.2.2. Network Switching Module. The module receives the network programming instruction table generated by the network control module and performs data packet processing and forwarding according to the table. It includes the following:

(i) Packet processor filters the data according to the matching rules in the instruction table, then splits, assembles, and adjusts the data packets according to the control commands, and sends them to the forwarding controller

(ii) Forwarding controller forwards the data according to the control commands and instruction table

(iii) Network programming instruction table is a command list consisting of forwarding and packet processing instructions, which includes data match description in each line. The forwarding and packet processing instruction is distributed by the network control module to the network switching module

6.2.3. Network Monitoring Module. The module is responsible for monitoring the status of the component, receiving and sending the statistics and exchanging information, querying the statistical commands sent by the network-aware programmable interface, and communicating between the network control module and network switching module. The monitoring information mainly includes the dataflow, quality and theoretical bandwidth of the channel, and resource occupation of each node of the channel.

6.2.4. Network-Aware Programming Interface. The interface drives the monitoring module to collect information such as the connection status, bandwidth, and dataflow in the network and calculates path bandwidth utilization and status threshold, analyses and optimizes the network path, locates failures, and implements network-aware programming.

6.2.5. Network Schedule Programming Interface. The computational task uses this interface to drive the network control module to generate, publish, and adjust the network programming instruction table according to the network status and the computing task, to realize the real-time dynamic scheduling of the network on demand.

The programmable protocol of DANET is used to regulate the transmission of data, control commands, and

monitoring awareness information; the red part is shown in Figure 18.

6.2.6. Instruction Table Distribution and Network Scheduling Protocol. The protocol is used between the network control module and the network switching module to transmit scheduling commands and programming instruction tables, which mainly include the following: collect, update, release, merge, and serialize the instruction table.

6.2.7. Packet Processing Protocol. The protocol is used between the network switching module and data network to transmit and standardize the processing mechanism for messages, including the definition of the structure of the data package format and the packaging/unpacking methods for data packets in the DANET.

6.2.8. Network State Awareness Protocol. The protocol is used between the network control module and the network monitoring module to collect network status, monitor commands, and collect network awareness information, including the network status collection command format, the monitoring data message definition format, the regulation of the flow of messages and commands, and the monitoring data aggregation.

7. Task Awareness and Computing Scheduling of DCOMP for the Tactical Edge

7.1. Discovery and Decomposition Task in the Tactical Edge for DCOMP. In the harsh battlefield environment, the mismatch between the resources of the computing nodes and the requirements of the combat mission will lead to the low utilization rate of the resources. The relationship mapping model between the attributes and the potential requirements of the task should be established, and the task group should be found according to the multiple attributes of the task, to provide the support for the computing in the PDCN.

In order to cope with the sudden increase in resource requirements for combat mission in the harsh battlefield environment of the future tactical edge, it is necessary to offload the computing tasks to the PDCNs in DANET for collaborative computing to reduce the load of a single computing node. The goal of the computation offloading in the nodes of DCOMP is to reduce the resource excessive consumption in a single computing node and the network load on the premise of ensuring the QoS of task quality in the tactical edge. To effectively solve the problem of resource competition and quality QoS degradation caused by the surge of computing tasks, the computation offloading of DCOMP should consider the participation of multiple computing nodes, multiparty collaboration, and multiple factors, which include computing, storage, and the cost of communication resources. In order to minimize the cost and maximize the benefit for DCOMP, it is necessary to build the task awareness and computing scheduling mechanism by considering the constraints of benefit and cost functions. On the premise of satisfying the constraint conditions, the benefit and cost of each computing node in the DANET will be calculated, which can maximize the benefit and performance of the whole calculation platform to complete the task.

Table 3: Combat mission attribute table.

Attributes	Central characteristic
Task time	Time or period to perform a task
Task location	Locations frequently performing operational task
Resources required	Resources frequently needed by task
Resource offset of task requirements	Task-specific operational resources needed

7.1.1. Relationship Mapping Model of Task Attributes and Potential Requirements. There is a relationship between the attributes of most combat tasks on the tactical edge and their potential resource requirements. Due to the large scale of combat tasks and attributes, it is necessary to build a model to simplify the mapping relationship between task attributes and potential requirements to facilitate the discovery of computing tasks with typical central characteristics in multiple dimensions. For example, the frequency of combat location and the specific times have typical central characteristics. Therefore, in addition to the conventional attributes of time, space, and priori knowledge, many can be added to design more behavioural attributes around the centrality of the combat missions. The centrality of the combat missions is shown in Table 3:

It can use the centrality discovery algorithm to obtain the centrality information of the combat mission under multiple attributes from several records and select the central offset or distance of multiple attributes to describe the tactical task.

Firstly, the information such as time period t and location l can be extracted, and then, calculate the center offset of each attribute according to the information. The original spatiotemporal information is projected into a new vector space $d_1(e_1)$, where the coordinates of each point represent the offset of the corresponding center point from it.

The mapping model of the potential requirements and the attributes of combat missions can be described as a nonlinear model, and the multidimensional attributes of combat mission access content, location, time, and traffic extracted at time t can be represented as

$$X_t = (x_1(t), x_2(t), x_3(t), \cdots, x_n(t)), \tag{8}$$

where $x_1(t), x_2(t), x_3(t), \cdots, x_n(t)$ represent multidimensional attributes, respectively.

The potential demand of the combat mission is expressed as the required resource of the combat mission at the next moment $t+1$, expressed as $Y(t)$. The importance of multiple attributes to the potential requirements of a combat mission is

$$A = (a_1, a_2, a_3, \cdots, a_n), \tag{9}$$

where $a_1, a_2, a_3, \cdots, a_n$ represent the importance of multidimensional attributes, respectively.

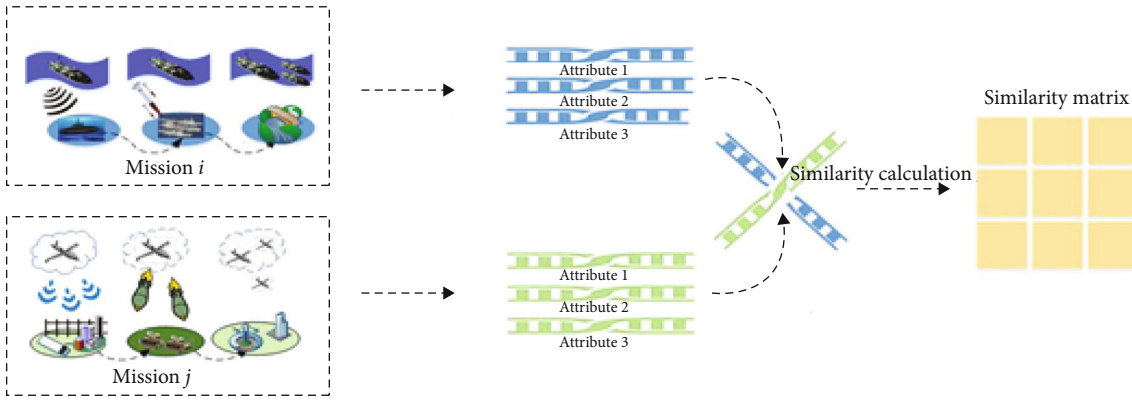


Figure 19: The calculation process of the time series similarity matrix.

The potential relationship between tactical tasks and multiattribute mapping can be described as

$$\begin{aligned} Y(t+1) &= f(a_1x_1(t), a_2x_2(t), \cdots, a_nx_n(t)), \\ Y(t+1) &= f\left((a_1, a_2, \cdots, a_n)^T \bullet (x_1(t), x_2(t), \cdots, x_n(t))\right), \\ Y(t+1) &= f\left(A^T \bullet X\right), \end{aligned} \tag{10}$$

where f is a nonlinear function, which can be approximated as

$$f(x) = \sum^{M} P_i(x) w_i \, (w_i \in R), \tag{11}$$

where $P_i(x)$ is a polynomial function that is generally used as a basis function in models for nonlinear system identification.

It can choose an entropy-based classification algorithm (random forest or decision tree algorithm) to build the relationship between multiple attributes and potential requirements of the combat mission, which can take $X(t) = (x_1(t), x_2(t), \cdots, x_n(t))$ as input and $Y(t+1)$ as output for model training.

The random forest algorithm or decision tree algorithm constructs the decision tree for the extracted training data set and feature subset, and the final classification result is classified by the decision tree voting, as shown in the following formula:

$$H(x) = \arg\left(\max_v \left(\sum^{k} (I(h_i(x) = Y))\right)\right), \tag{12}$$

where $H(x)$ is a combined classification model, h_i is a single decision tree, I is the characteristic function, and Y is the target variable. The split indicator of the decision tree is Gini. The importance of each attribute can be obtained by calculating the mean of the Gini index.

7.1.2. Large-Scale Efficient Similarity Matrix Calculation. The time, location, access content sequence, and traffic of the task attributes with different dimensions and different value ranges need to eliminate the differences between the attributes and characterize the dynamic behaviours. By calculating the similarity of the specific attribute time series of different tasks and eliminating the difference in attribute dimensions, the similarity of the time series containing the dynamics of the combat mission can be obtained to form a similar matrix of task behaviour.

The similarity matrix of combat mission of time series is used to form the similarity matrix between combat missions. Figure 19 shows the calculation process of the time series similarity matrix for combat missions.

The attribute sequence A_i and A_j of the combat mission is extracted from the temporal composition sequence of the combat mission. According to the attribute sequence of the combat mission, the appropriate similarity calculation method can be used to calculate the similarity P_{ij} between the two combat missions. For all combat missions, the similarity under different attributes is calculated in pairs, and the similarity matrix P_S and P_T under multiple attributes of the combat mission can be obtained. P_S and P_T are the space location and time of the combat mission, respectively:

$$\begin{aligned} P_S &= \begin{bmatrix} p_S^{11} & \cdots & p_S^{1m} \\ \vdots & \ddots & \vdots \\ p_S^{m1} & \cdots & p_S^{mm} \end{bmatrix}, \\ P_T &= \begin{bmatrix} p_T^{11} & \cdots & p_T^{1n} \\ \vdots & \ddots & \vdots \\ p_T^{n1} & \cdots & p_T^{nn} \end{bmatrix}. \end{aligned} \tag{13}$$

It is assumed that similar combat missions in geographical location will be more similar, while those far away in geographical location will be less similar. According to the geographical position of the combat mission, the combat mission can be divided to calculate the similar matrix preferentially similar in geographical position, which can greatly reduce the amount of calculation of the similarity matrix. The specific algorithm is shown in Figure 20.

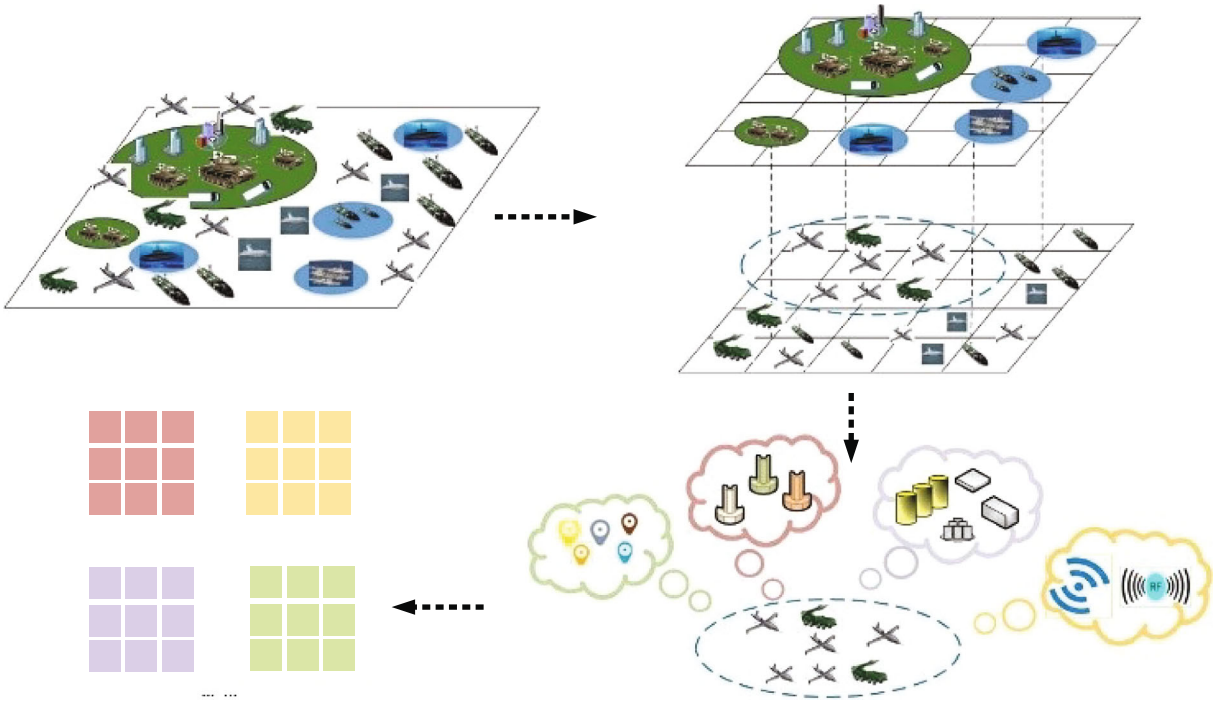


Figure 20: Large-scale efficient similarity matrix calculation method.

7.1.3. Resource Discovery Based on Weighted Similarity Matrix Fusion. After obtaining the mapping model of combat mission attributes and potential resource requirements, in order to eliminate the parameter dimension differences between the attributes and quickly find the combat mission group, it is necessary to fuse the similarity matrices of multiple attributes and segment the fused similarity matrix.

The similarity matrix fusion method can effectively improve the system's resolution ability by fusing similarity matrices with different attributes, while retaining the information of the original similarity matrix to the greatest extent. Therefore, the sum of the distance between the fused similarity matrix and the original similarity matrix should be the smallest, as shown in Figure 21.

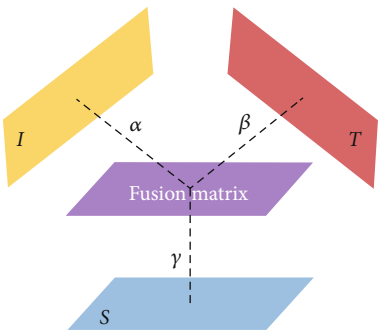


Figure 21: Schematic diagram of fusion similarity matrix calculation.

Set P to be the similarity matrix after fusion, the weights α, β, γ reflect the behaviour attribute, and S, T, I reflect the importance of the potential needs of tactical task. The goal of calculation of the similarity matrix is to minimize the sum of the distances between P and the original similarity matrix P_S, P_T, P_I of multiple attributes. The specific model is described as follows:

$$\min \alpha\|P - P_I\|_2^2 + \beta\|P - P_T\|_2^2 + \gamma\|P - P_S\|_2^2 + \lambda\|P\|_1, \quad (14)$$

$$\text{s.t.}\alpha + \beta + \gamma = 1, \quad (15)$$

$$\alpha \geq 0, \beta \geq 0, \gamma \geq 0, \quad (16)$$

$$1 \geq P_{ij} \geq 0, i, j \in [1, N]. \quad (17)$$

The second norm term in formula (14) indicates the distance of the similarity matrix of the corresponding data relative to the fusion result matrix, and λ is the sparse rule operator. Formula (15) represents the normalized weights of the three types of data relative to the final fusion result. Formula (16) limits the weights of various types of data in the optimization formula and the range of sparse rule operator weights. Formula (17) shows the value range of each element in the fusion result matrix P. $\lambda\|P\|_1$ is a continuous convex function, and $\alpha\|P - P_I\|_2^2 + \beta\|P - P_T\|_2^2$ is a continuously differentiable function and satisfies the Lipschitz conditions. The Lipschitz continuous gradient is $L(f)$. Set $f(P) = \alpha\|P - P_I\|_2^2 + \beta\|P - P_T\|_2^2$ and $g(s) = \lambda\|P\|_1$, then

$$\left\|\nabla f\left(P'\right) - \nabla f(P)\right\|_2^2 \leq L(f)\left\|P' - P\right\|_2^2. \quad (18)$$

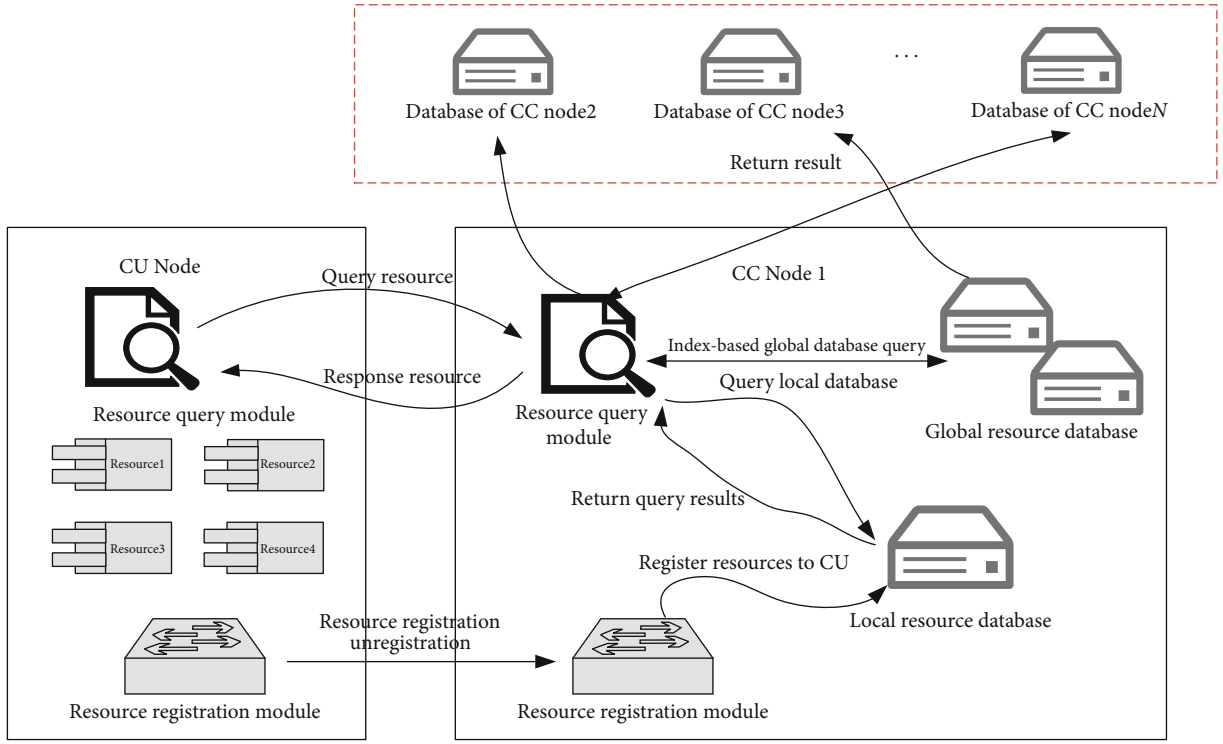


Figure 22: Structure diagram of a resource-aware process.

For such problems, one can use a fast-iterative shrinkage-thread should algorithm (FISTA) for calculation at S as follows:

$$P(S) + < \nabla f(S), S' - S > + \frac{L}{2}\left\|S' - S\right\|^2 = \frac{L}{2}\left\|S' - S + \frac{1}{L}\nabla f(S)\right\|^2 + \text{const}, \tag{19}$$

where L is the Lipschitz constant, and the available iteration value by ignoring the constants is

$$S_{k+1} = \text{argmin}_s \left\{\frac{L}{2}\left\|S - \left(S_k - \frac{1}{L}\nabla f(S_k)\right)\right\|_2^2 + \lambda\|S\|_1\right\}. \tag{20}$$

The initial value is given according to the actual situation during the calculation and iterates according to the above formula until convergence, to obtain the final similarity matrix after fusion.

In the process of calculation and fusion of similar matrices, various attributes of combat missions have been included and the information of requirements of potential combat mission and activity locations can be obtained, providing resource scheduling and matching for DCOMP.

7.2. Real-Time Perception and Management of Resources for DCOMP. At the tactical edge, various wireless links are widely used and caused the different transmission qualities. Obviously, the DANET is a heterogeneous network with a mixed network of broadband and narrowband in a complex battlefield environment, which can be subject to various kinds of aggression such as battle and electronic interference impact. In DANET, the resource management is a dynamic process that can be dynamically perceived and discovered. The resources mainly include various resources of different network bandwidths, services, and hardware in DANET. As the topology of DANET changes dynamically, the resource-aware algorithm should adapt to the plug and play of available resources, and the resources of nodes can be perceived in time by other nodes in the entire network of DCOMP. The purpose of resource awareness is when a resource is provided by a PDCN in DANET, other nodes in the network can identify the resource, and when the resource is revoked, other nodes can also know that the resource is out of date and no longer provided.

By considering the limitations of nodes on memory, CPU, storage, and other resources, the algorithm running

Table 4: Data package structure of resource description information.

Serial number	Packet definition	Message description
1	ResName	Name of the resource
2	ResId	Category ID of the resource
3	ResHostId	Host address of the resource provider
4	ResExpTime	Expiration time of the resource
5	ResPosition	Location of the resource provider
6	ResDes	Description of the resource function
7	ResQuality	Resource quality

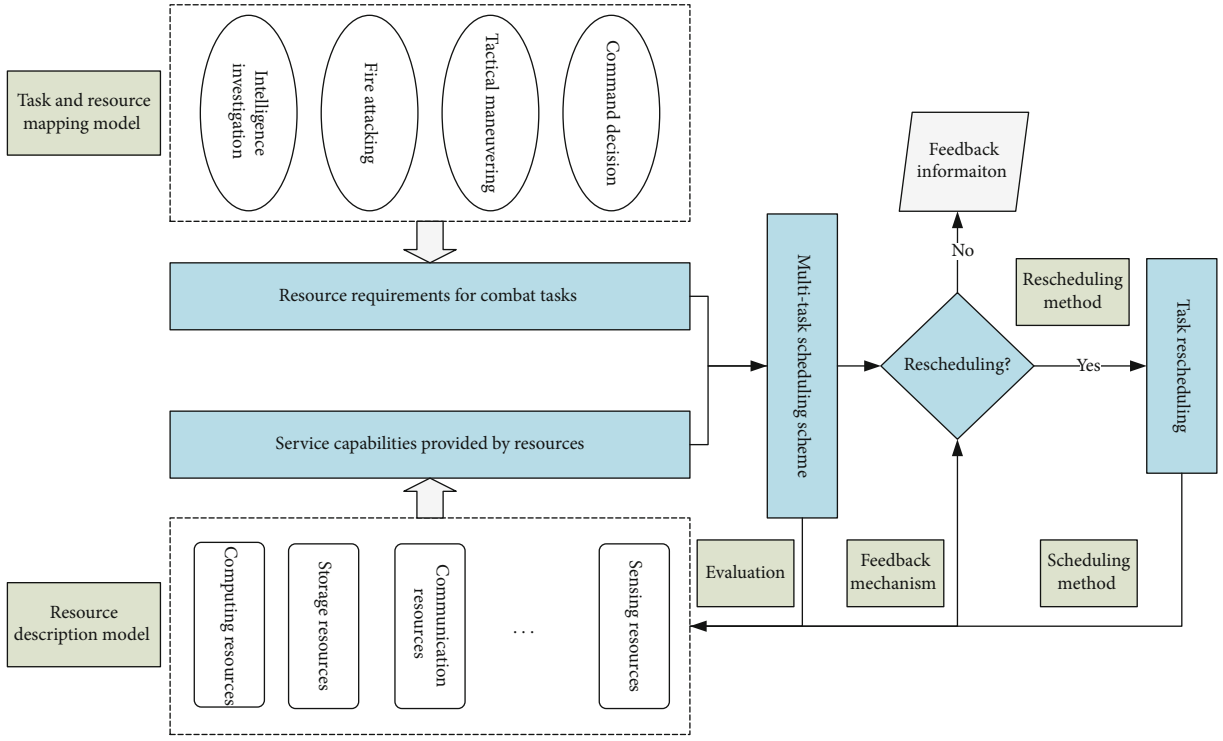


Figure 23: Task-coordinated planning and scheduling mechanism.

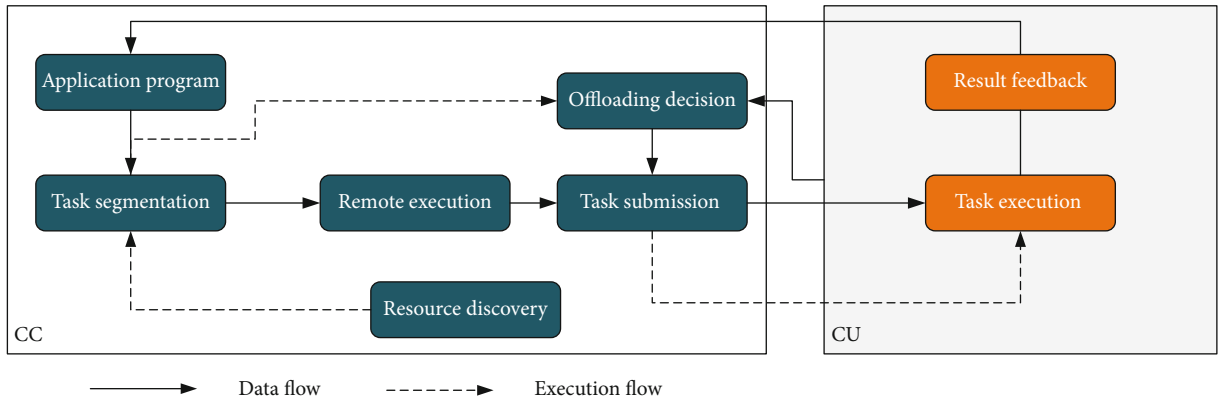


Figure 24: The workflow of computation offloading.

on PDCN should not be too complicated. According to the different statuses of nodes in DANET, the protocol can be divided into resource awareness modules, client agents on the common node (CU node), and directory agents on cluster head nodes (CC node). Figure 22 shows the structure diagram of a resource-aware process.

As shown in Figure 22, the CU node is generally responsible for the calculation of various tasks, which needs to register its own fixed resources (camera, storage, communication resources, and various sensors) to the CC node through the resource registration module. Various real-time changing resources (real-time CPU/memory usage, network bandwidth) can be queried/responded by the CC node through the resource query module. The CC node is a cluster head node responsible for managing the resources of the CU nodes and collaborative communicating and computing with external CC nodes. Generally, a CC node has a local resource database, a resource registration module, a resource query module, and a global resource database. The resource registration module obtains the resource data of the CU nodes within the range of the CC node and stores in the local resource database. A CC node regularly manages and backs up the computing resource data of the entire DANET by collecting resource data from other CC nodes. Other CC nodes also can obtain the resource data from the resource query mode of this CC node. The resource database of the CC node mainly stores various resource description data including the name, node address, and some related descriptions of the resource. The resource description information in the traditional SDP (Service Discovery Protocol) only contains the

resource name and the address of the resource provider [80]. The resource description of DANET should contain the information such as function description, resource category, provider identification, network environment parameters, performance state parameters, and location. Therefore, when multiple resources can meet task needs in terms of functions, the optimal node can be selected based on the information to perform the computing task. For a DANET, resource description information needs to include the following aspects: name, category ID, host address, expiration time, location, resource function description, and quality information of the resource. Therefore, resource description information is the basis of the resource aware protocol, and the carrier of resource description information transmission in the protocol is the definition of the data package. Table 4 shows the data package structure of resource description information.

7.3. Resource Planning and Scheduling for DCOMP. In the complex and changeable battlefield environment, according to the objective, determination, and mission of the combat mission, starting from the integration support of information acquisition, integrated processing, and strike command communication, it is necessary to perform reasonable metatask scheduling according to the status and resource capabilities of DANET and achieve reasonable sharing of resources and mutual cooperation for completing the combat mission. The scheduling problem of computing resources is a mixed constraint planning problem that involves both time and resource allocation. Based on the resource description, task, and resource mapping model, the formal definition is established from the ability of resources to meet mission requirements and the priorities of the defined process and integrity constraints.

In order to realize the dynamic optimal scheduling and conflict avoidance of computing tasks and resources, reduce the time complexity of the whole network resource collaborative processing and load of resource providers and improve the efficiency and robustness of DCOMP, to achieve the goal of global optimization and comprehensive combat mission execution successfully. The collaborative planning and scheduling of computing tasks are shown in Figure 23.

Computing task planning and scheduling are determined based on the characteristics of combat mission requirements and the resource characteristics. Different computing tasks have different requirements for resources, and different resources have different capabilities to meet the task, which leads to multitask planning requirements. In multitasking planning and scheduling, the satisfaction of task requirements and resource consumption determine the comprehensive efficiency of a multitasking system. The satisfaction of task requirements has a positive impact on comprehensive efficiency, and excessive resource consumption has a negative impact. According to the parameters of computing task and resource provider, it is necessary to define planning objectives and related constraints and establish a multitasking planning model. According to the comprehensive efficiency and its changing rule of "task number" and "resource number" in different application scenarios, the optimal ratio of "task number" and "resource number" can be obtained, and the relative optimal multitasking arrangement scheme needs to be established.

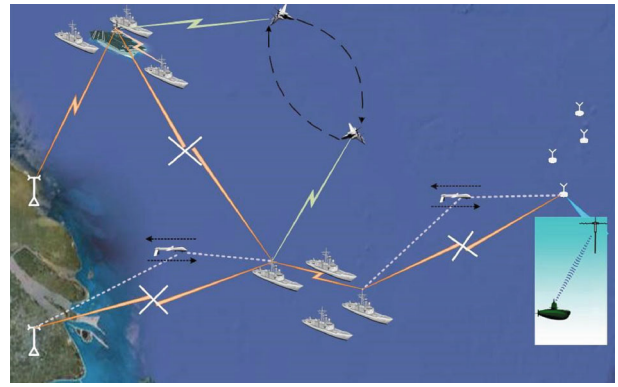
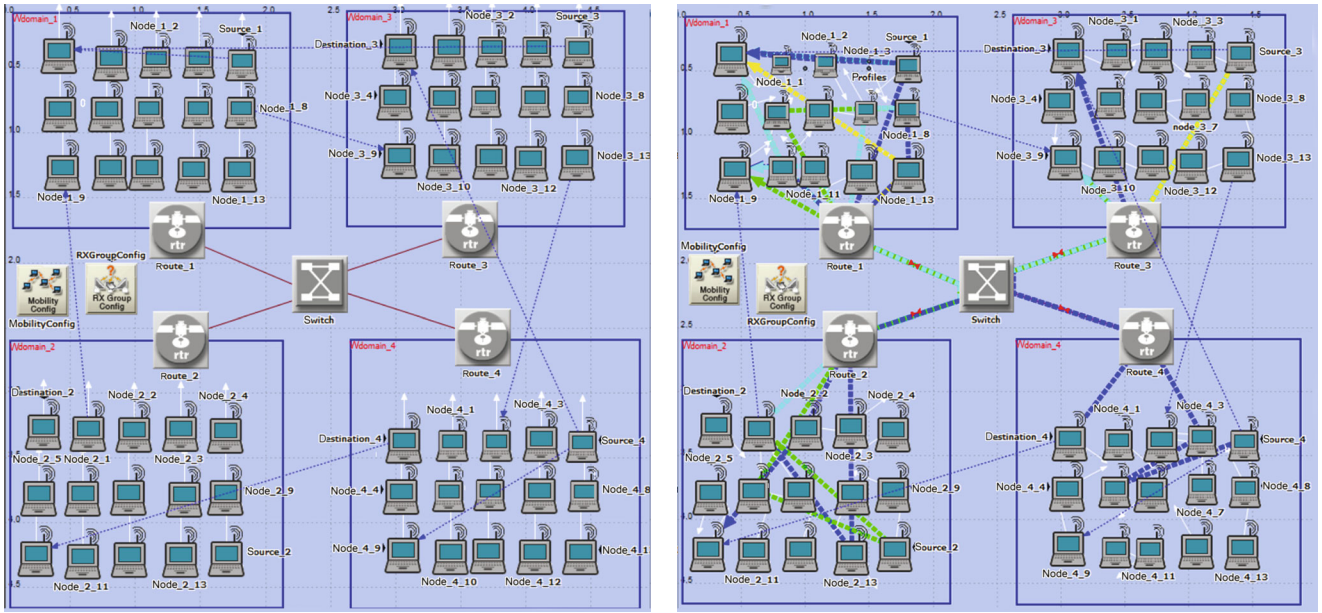


Figure 25: A typical tactical edge combat scenario.

In the actual task execution process, the possible problems such as adding, deleting, and changing tasks and running faults may occur frequently, and a new task scheduling plan needs to be recalculated according to the changes and the original task planning.

7.4. Computation Offloading for DCOMP. The calculation, storage, and battery capacity of a single computing node is very limited, but the tasks that need to be processed on the node are becoming more and more complicated in the tactical edge. There is a contradiction between the large amount of computing resources consumed by computing intensive applications and the limited resources of a single computing node. In order to solve the contradiction between the limited service resources and the unlimited computing task requirements, the task needs to be distributed to the node at DANET for computational offloading [81]. Computation offloading is mainly performed on the CC nodes, which can offload some computing tasks to the CU nodes on DANET. The nodes performing computation offloading not only need to send computing tasks and receive computational results but also need to execute computing tasks, to reduce response delays for time-intensive tasks, which is different from the traditional computing paradigm of MEC, FC, etc. The major steps include computing node discovery, task segmentation, offloading decision, task submission, remote task execution, and calculation result feedback. The traditional computation offloading uses virtual machine migration to deploy the entire application to the service platform for execution, requires high network bandwidth, and results in low efficiency, which is not suitable for the environments of complex, highly dynamic, and weakly connected at tactical edges which is not suitable for complex and highly dynamic environments, leading to weak connection at tactical edges. [82]. The process of computation offloading is shown in Figure 24.

(i) *Resource Discovery*. Find collaborative computing nodes that can perform computing tasks in the current DANET. The computing node can be a high-performance computer located in a remote



(a) Network model (b) Routing path

Figure 26: The OPNET network model of DANET.

data center or a handheld terminal with limited computing ability

(ii) *Task Segmentation*. In the preparation phase of computation offloading, the results of the segment of computational tasks have a significant impact on the performance of computation offloading. The granularity of task segmentation can be divided into method, module, and thread levels based on the discovered resources

(iii) *Offloading Decision*. Deciding whether to perform computation offloading to a CU node mainly depends on the overhead of network delay, the energy consumption, the current status of the CU node, and the computing task

(iv) *Remote Execution and Task Submission*. This module is responsible for packaging the programmable code (described in Section 4) and data that needed to be calculated and sent to the CU node

(v) *Task Execution*: Execute the programmable code offloaded to the CU node according to the programmable model described in Section 5

(vi) *Result Feedback*. The calculation result feedback is the last step of the computation offloading. After the CU node feeds back the calculation results to the CC node, the network connection between the CU node and the CC node is released and the computation offloading is finished

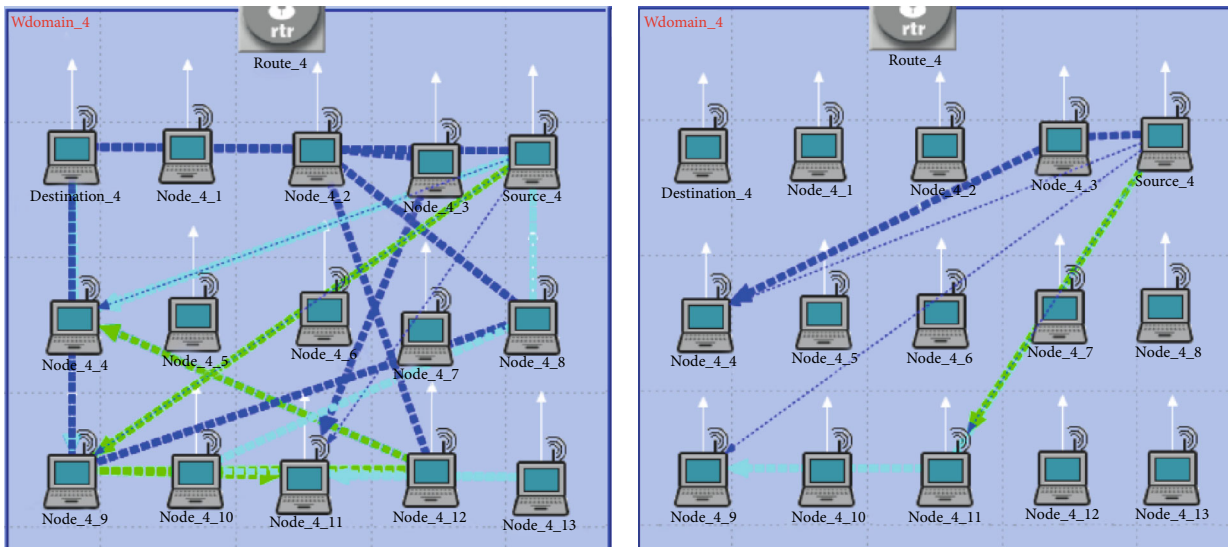
All the resources of computing nodes are deployed in a distributed manner on the DANET, and each computing node can upload calculation results anytime and anywhere and at any movement speed [83].

8. The Application Scenario Analysis of DCOMP in Future Wars

In the future, military operations such as evacuation, peacekeeping, and counter-terrorism will be far away from the homeland, and the operation process will lack the support of infrastructure with strong communication and data processing capabilities, which need to rely on a network that can handle dynamic tasks. The traditional computing paradigms such as cloud computing, edge computing, and wireless sensor networks cannot satisfy this kind of far way form command center, which network throughput is severely limited and the battlefield situation changes rapidly.

Figure 25 shows a typical battle scenario of the tactical edge away from the homeland, which consists of aircraft carriers, frigates, destroyers, fighter jets, UAVs, etc. The UAVs are used for reconnaissance, fighter jets for strikes, aircraft carriers for commanding operations, and other warships for escorts.

For example, when an UAV carries out reconnaissance, the images of the enemy are collected, which need to be analysed and processed. At this time, the data link received enemy interference and the communication between the UAV and the carrier was interrupted. However, the communication link between UAVs and fighter jets is still available. The DANET can be immediately established between the UAVs and fighter jets, which will distribute the collected image and offload the image processing program to the PDCN for calculation, and the calculated results will be sent to the UAV or other combat units for decision-making. The role of DCOMP is to quickly organize these combat units due to the lack of computing infrastructure support to conduct various real-time task calculations at the tactical edge. Therefore, DCOMP will play a very import role in future wars, and the large-scale application of DCOMP will change the way



(a) Traditional protocol (b) Improved protocol

Figure 27: Comparison of routing calculation results between the traditional and improved protocols.

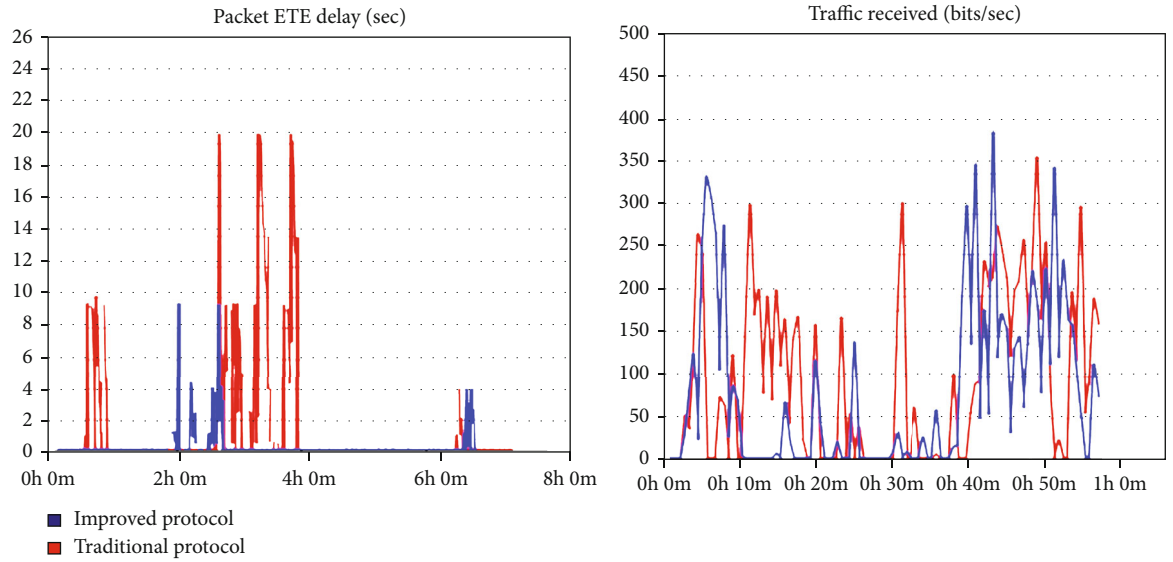


Figure 28: Simulation result of packet ETE delay and traffic received.

and even the course of warfare. In order to verify the performance of the DANET proposed in this paper, we designed a network model to simulate the communication progress of DCOMP at the tactical environment by OPNET network simulation software, as shown in Figure 26.

The network model constructed in Figure 26(a) contains 4 subnets with 15 PDCNs and a router using the AODV or OLSR routing protocol connected by a switch. The PDCN moves within a certain range at the tactical edge to achieve collaborative computing and data sharing with DANET. Since the AODV protocol is an on-demand protocol, when creating a routing table, different time sequences for service requests can be considered and different transmission paths can be established. As can be seen from Figure 26(b), when the existing link in the network is under heavy load and receive the routing request again, the link load status of the existing traffic should be considered. Therefore, the routing protocol should avoid the overloaded link and choose the "idle" or "heavy" path as much as possible to forward data packets. Figure 27 shows the simulation result of the routing path in a subnet using the traditional protocol and the improved protocol algorithm proposed in this paper.

As can be seen from Figure 27, it can be easily found that with the standard protocol, the link load is already heavy and there are still multiple routing requests whose final data forwarding still passes through the link, which will easily cause the loss of a large number of data packets and cause large delay of the network. The statistics of package end-to-end delay and traffic received between the traditional and improved protocols is shown in Figure 28.

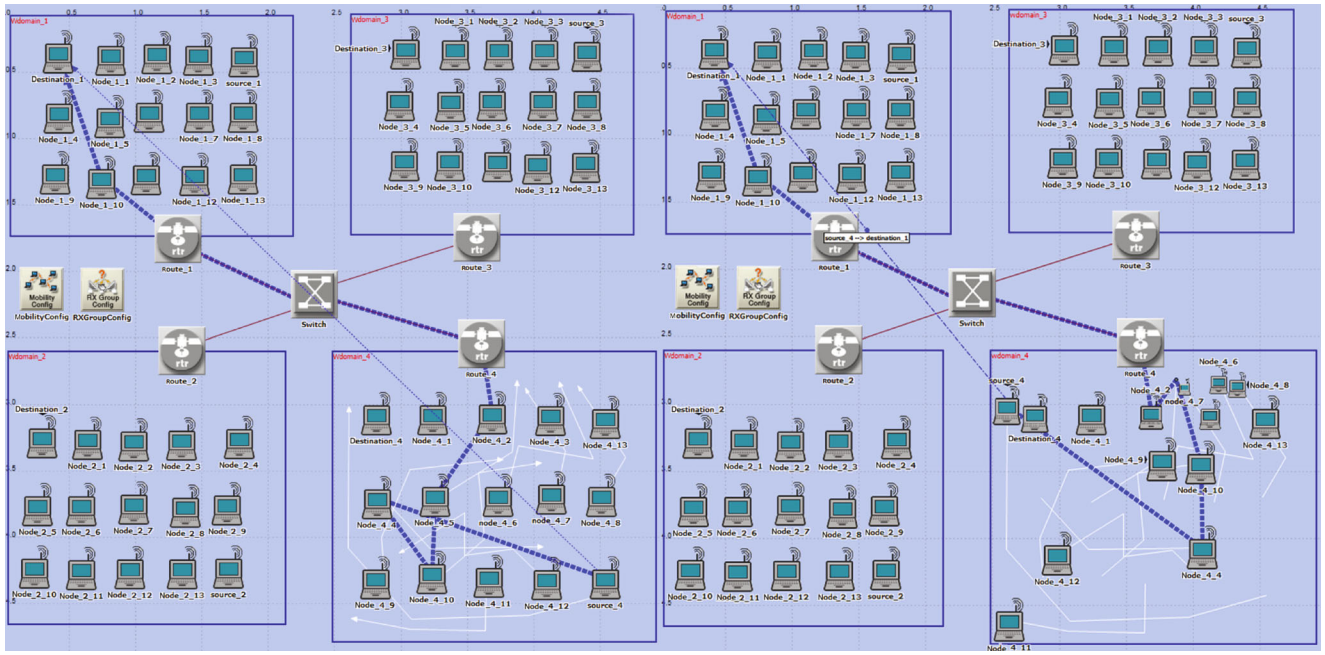


Figure 29: Simulation result of fast movement of the computing node.

As can be seen from Figure 28, the packet ETE delay of the improved algorithm is much lower than that of the traditional algorithm and the package of traffic received is not much difference.

At the tactical edge, the speed and path of each computing node are different, which has a great impact on the selection of network routing, throughput, and packet loss rate. With the movement of nodes, the nodes that could not be connected may become connected due to the change of relative position and the topology of the whole network will change with the movement of each node. As shown in Figure 29, which is the schematic diagram of network topology after simulation for 0 s and 30s, with the improved AODV protocol, the link cost between nodes will be changed due to the change of topology structure, and the routing path of services will also change accordingly.

It can be seen from the simulation that the improved protocol can greatly reduce the network delay and improve the network performance, providing a good network environment support for DCOMP. Further, we will perform a large-scale experiment of DCOMP to substantiate our findings and strive to be able to be applied in practical engineering and on a real environment.

9. Conclusion and Challenges in the Future

The rapid transformation of warfare, the continuous upgrading of weapons, and the diversity of combat missions have led to the continuous change of the battlefield network environment, and new requirements have been imposed on environmental elements such as computing, network, and command. We investigated in detail a range of the main technical concepts, mechanisms, paradigms, and important features of DCOMP to meet the requirements and development trends of future battlefield computing and communications. In this article, we have proposed the architecture for DCOMP, a network model called DANET, and the programming model and language for DCOMP. We have also presented methods of task awareness and computing scheduling for the tactical edge in a harsh battlefield environment. Moreover, we have discussed a typical scenario and a vision of DCOMP for the tactical edge in future wars.

As a new computing paradigm, the research on DCOMP in the academia has just started, and the DCOMP has certain advantages over the traditional computing paradigm such as MCC, FC, and MEC. However, limited by the complicated tactical edge network environment which caused discontinuous communication between devices, simulation in real scenes is somewhat difficult and complex. There are still many problems and challenges that need to be studied in the future:

(i) The development of highly dynamic and programmable network protocols for DCOMP should be accelerated as soon as possible to enable it to adapt to weakly connected, highly dynamic, and vulnerable environments at the tactical edge to win the future wars

(ii) Accelerate the research on the technology of programmable computing nodes, programmable networks, and computation offloading. According to different application requirements and network conditions, it is possible to develop a migration solution for nodes adapted to various types of equipment and networks in DCOMP. In addition, it is necessary to establish relevant stands and upgrade existing equipment to meet the requirements of DCOMP

(iii) Many current applications used in the tactical edge should be modified to be adapted for DCOMP, such as tactical target recognition and tracking, target damage analysis, and trajectory analysis

(iv) It is a great challenge to find the right balance between data transmission and computation resources in the degenerate network, which requires in-depth study of the optimal control strategy between real-time network state, dispersed computing resources, and computing tasks

(v) How to ensure the quality of service (Qos), the efficiency of computation, and the security of data is the key research direction in the future under the condition of limited and variable bandwidth and highly dispersed heterogeneous computing resources

(vi) Coded computing is a recent technique that will enable optimal trade-offs between computation load, communication load, and computation latency due to stragglers in DCOMP [63], designing joint task scheduling and coded computing in order to leverage trade-offs between computation, communication, and latency, which is an important aspect in DCOMP

The tactical edge is generally far from the homeland, the nodes of DCOMP are generally powered by batteries, and the battery life is limited. Therefore, the technology for energy consumption management of DCOMP will also need to be studied as a key direction in the future. In the future, the technology of DCOMP can be applied not only in military fields at the tactical edge but also in civil fields such as fire rescue, flood relief, and environmental monitoring.

Data Availability

All data, models, and codes generated or used during the study appear in the submitted article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the National Defense Basic Scientific Research Program of China under Grants JCKY2019210B005, JCKY2018204B025, and JCKY2017204 B011, the Key Scientific Project Program of National Defense of China under Grant ZQ2019D20401, the Chinese Defense Advanced Research Program of Science and Technology under Grant 41401030301, and the Open Program of National Engineering Laboratory for Modeling and Emulation in E-Government (item number MEL-20-02).

References

[1] L. Yun-Tung, "A unified service description for the global information grid," *CrossTalk*, vol. 20, no. 4, pp. 23–26, 2007.

[2] Z. Guo-Hong, "Architecture of combat clouds," *Journal of Command and Control*, vol. 1, no. 3, pp. 292–295, 2015.

[3] J. George, C. Chen, R. Stoleru, G. G. Xie, T. Sookoor, and D. Bruno, "Hadoop MapReduce for tactical clouds," in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pp. 320–326, Luxembourg, Luxembourg, 2014.

[4] A. E. Conway, M. Wang, E. Ljuca, and P. D. Lebling, "A Dynamic Transport Overlay System for Mission-Oriented Dispersed Computing Over IoBT," in *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, pp. 815–820, Norfolk, VA, USA, 2019.

[5] M. R. Schurgot, M. Wang, A. E. Conway, L. G. Greenwald, and P. D. Lebling, "A dispersed computing architecture for resource-centric computation and communication," *IEEE Communications Magazine*, vol. 57, no. 7, pp. 13–19, 2019.

[6] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 319–333, 2019.

[7] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[8] M. Jo, T. Maksymyuk, B. Strykhalyuk, and C.-H. Cho, "Device-to-device-based heterogeneous radio access network architecture for mobile cloud computing," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 50–58, 2015.

[9] D. Han, W. Chen, B. Bai, and Y. Fang, "Offloading optimization and bottleneck analysis for mobile cloud computing," *IEEE Transactions on Communications*, vol. 67, no. 9, pp. 6153–6167, 2019.

[10] Y. Miao, G. Wu, M. Li, A. Ghoneim, M. Al-Rakhami, and M. S. Hossain, "Intelligent task prediction and computation offloading based on mobile-edge cloud computing," *Future Generation Computer Systems*, vol. 102, pp. 925–931, 2020.

[11] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: a survey, state of art and future directions," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 133–143, 2014.

[12] D. Tychalas and H. Karatza, "A scheduling algorithm for a fog computing system with bag-of-tasks jobs: simulation and performance evaluation," *Simulation Modelling Practice and Theory*, vol. 98, article 101982, 2020.

[13] A. V. Dastjerdi and R. Buyya, "Fog computing: helping the Internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[14] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[15] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, 2016.

[16] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: fog, cloudlet, mobile edge, and micro data centers," *Computer Networks*, vol. 130, pp. 94–120, 2018.

[17] X. Gao, X. Huang, S. Bian, Z. Shao, and Y. Yang, "PORA: predictive offloading and resource allocation in dynamic fog computing systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 72–87, 2020.

[18] Q. Wu, H. Ge, H. Liu, Q. Fan, Z. Li, and Z. Wang, "A task offloading scheme in vehicular fog and cloud computing system," *IEEE Access*, vol. 8, pp. 1173–1184, 2020.

[19] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3702–3712, 2016.

[20] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, 2017.

[21] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, and M. Dlodlo, "From cloud to fog computing: a review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[22] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[23] H. Elazhary, "Internet of things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions," *Journal of Network and Computer Applications*, vol. 128, pp. 105–140, 2019.

[24] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[25] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[26] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[27] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[28] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, 2017.

[29] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[30] A. Zhou, S. Wang, S. Wan, and L. Qi, "LMM: latency-aware micro-service mashup in mobile edge computing environment," *Neural Computing and Applications*, vol. 32, no. 19, pp. 15411–15425, 2020.

[31] Y. Cao, H. Song, O. Kaiwartya et al., "Mobile edge computing for big-data-enabled electric vehicle charging," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 150–156, 2018.

[32] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[33] A. K. Malhi, S. Batra, and H. S. Pannu, "Security of vehicular ad-hoc networks: a comprehensive survey," *Computers & Security*, vol. 89, article 101664, 2020.

[34] W. Kiess and M. Mauve, "A survey on real-world implementations of mobile ad-hoc networks," *Ad Hoc Networks*, vol. 5, no. 3, pp. 324–339, 2007.

[35] L. Hanzo and R. Tafazolli, "A survey of QoS routing solutions for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 2, pp. 50–70, 2007.

[36] A. Ghaffari, "Hybrid opportunistic and position-based routing protocol in vehicular ad hoc networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1593–1603, 2020.

[37] F. Mari, I. Melatti, E. Tronci, and A. Finzi, "A multi-hop advertising discovery and delivering protocol for multi administrative domain MANET," *Mobile Information Systems*, vol. 9, no. 3, pp. 261–280, 2013.

[38] P. Millán, C. Aliagas, C. Molina, R. Meseguer, S. F. Ochoa, and R. M. Santos, "Predicting topology propagation messages in mobile ad hoc networks: the value of history," *Sensors*, vol. 20, no. 1, p. 24, 2020.

[39] N. Bouchama, D. Aïssani, N. Djellab, and N. Nouali-Taboudjemat, "A critical review of quality of service models in mobile ad hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 21, no. 1, pp. 49–70, 2019.

[40] D.-G. Zhang, P.-Z. Zhao, Y.-y. Cui, L. Chen, T. Zhang, and H. Wu, "A new method of mobile ad hoc network routing based on greed forwarding improvement strategy," *IEEE Access*, vol. 7, pp. 158514–158524, 2019.

[41] S. A. Sharifi and S. M. Babamir, "The clustering algorithm for efficient energy management in mobile ad-hoc networks," *Computer Networks*, vol. 166, article 106983, 2020.

[42] Z. Jipeng, L. Liangwen, and H. Tan, "Traffic-predictive QoS on-demand routing for multi-channel mobile ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 2018.

[43] S. D. Ubarhande, D. D. Doye, and P. S. Nalwade, "A secure path selection scheme for mobile ad hoc network," *Wireless Personal Communications*, vol. 97, no. 2, pp. 2087–2096, 2017.

[44] M. R. Khosravi, H. Basri, and H. Rostami, "Efficient routing for dense UWSNs with high-speed mobile nodes using spherical divisions," *The Journal of Supercomputing*, vol. 74, no. 2, pp. 696–716, 2018.

[45] A. Pal and A. Jolfaei, "On the lifetime of asynchronous software-defined wireless sensor networks," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6069–6077, 2020.

[46] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.

[47] M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 85–96, 2014.

[48] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, pp. 24–37, 2006.

[49] Y. Li, S. Haiying, C. Kang, and L. Guoxin, "MobileCopy: Improving data availability and file search efficiency in delay tolerant networks against correlated node failure," *IEEE*

Transactions on Mobile Computing, vol. 20, no. 1, pp. 188–203, 2021.

[50] "Disruption tolerant networking (DTN)," 2008, http://www.darpa.mil/sto/solicitations/DTN/index.html.

[51] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 828–836, 2008.

[52] T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: taxonomy and design," *Wireless Networks*, vol. 16, no. 8, pp. 2349–2370, 2010.

[53] S. M. Tornell, C. T. Calafate, J.-C. Cano, and P. Manzoni, "DTN protocols for vehicular networks: an application oriented overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 868–887, 2015.

[54] M. Quwaider and S. Biswas, "DTN routing in body sensor networks with dynamic postural partitioning," *Ad Hoc Networks*, vol. 8, no. 8, pp. 824–841, 2010.

[55] N. Banerjee, M. D. Corner, and B. N. Levine, "Design and field experimentation of an energy-efficient architecture for DTN throwboxes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 554–567, 2010.

[56] M. Seligman, K. Fall, and P. Mundur, "Storage routing for DTN congestion control," *Wireless Communications and Mobile Computing*, vol. 7, no. 10, pp. 1183–1196, 2007.

[57] J. Whitbeck and V. Conan, "HYMAD: hybrid DTN-MANET routing for dense and highly dynamic wireless networks," *Computer Communications*, vol. 33, no. 13, pp. 1483–1492, 2010.

[58] Y. Mao, C. Zhou, Y. Ling, and J. Lloret, "An optimized probabilistic delay tolerant network (DTN) routing protocol based on scheduling mechanism for Internet of things (IoT)," *Sensors*, vol. 19, no. 2, p. 243, 2019.

[59] S. Wan, X. Xu, T. Wang, and Z. Gu, "An intelligent video analysis method for abnormal event detection in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2020.

[60] M. García-Valls, A. Dubey, and V. Botti, "Introducing the new paradigm of social dispersed computing: applications, technologies and challenges," *Journal of Systems Architecture*, vol. 91, pp. 83–102, 2018.

[61] D. Hu and B. Krishnamachari, "Throughput optimized scheduler for dispersed computing systems," in *2019 7th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pp. 76–84, Newark, CA, USA, May 2019.

[62] K. Fujikawa, H. Harai, M. Ohmori, and M. Ohta, "Quickly converging renumbering in network with hierarchical link-state routing protocol," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 6, pp. 1553–1562, 2016.

[63] C.-S. Yang, R. Pedarsani, and A. S. Avestimehr, "Communication-aware scheduling of serial tasks for dispersed computing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1330–1343, 2019.

[64] A. Knezevic, Q. Nguyen, J. A. Tran et al., "CIRCE-A runtime scheduler for DAG-based dispersed computing," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pp. 12–14, San Jose, CA, USA, October 2017.

[65] Q. Nguyen, P. Ghosh, and B. Krishnamachari, "End-to-end network performance monitoring for dispersed computing," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, pp. 707–771, Maui, HI, USA, March 2018.

[66] P. Ghosh, Q. Nguyen, and B. Krishnamachari, "Container orchestration for dispersed computing," in *Proceedings of the 5th International Workshop on Container Technologies and Container Clouds - WOC '19*, pp. 19–24, Davis, CA, USA, December 2019.

[67] J. Spillner and A. Schill, "Towards dispersed cloud computing," in *May 2014 in 2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 170–174, Odessa, Ukraine, May 2014.

[68] K. S. Meena and T. Vasanthi, "Reliability design for a MANET with cluster-head gateway routing protocol," *Communications in Statistics - Theory and Methods*, vol. 45, no. 13, pp. 3904–3918, 2016.

[69] Z. Lv, H. Song, P. Basanta-Val, A. Steed, and M. Jo, "Next-generation big data analytics: state of the art, challenges, and future research topics," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1891–1899, 2017.

[70] S. K. Goudos, P. D. Diamantoulakis, and G. K. Karagiannidis, "Multi-objective optimization in 5G wireless networks with massive MIMO," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2346–2349, 2018.

[71] 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.

[72] D. Anveshini, "LANMAR routing protocol to support real-time communications in MANETs using soft computing technique," in *Data Engineering and Communication Technology*, vol. 1079 of Advances in Intelligent Systems and Computing, pp. 231–243, 2020.

[73] H. Mohamed, M. H. Lee, M. Sarahintu, S. Salleh, and B. Sanugi, "Application of Taguchi's design of experiment in performance analysis of destination sequence distance vector (DSDV) routing protocol in mobile ad hoc networks," *Sains Malaysiana*, vol. 38, no. 3, pp. 423–428, 2009.

[74] D. Z. Rodríguez, R. L. Rosa, and P. H. M. de Lima, "New cache system-based power-aware algorithm in MANET," in *2010 Fifth International Conference on Digital Telecommunications*, pp. 86–91, Athens, Greece, June 2010.

[75] J. Toutouh, J. Garcia-Nieto, and E. Alba, "Intelligent OLSR routing protocol optimization for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1884–1894, 2012.

[76] R. Bai and M. Singhal, "DOA: DSR over AODV routing for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 10, pp. 1403–1416, 2006.

[77] R. B. Anand and P. Kumar, "Process automation of simulation using Toolkit/Tool Command Language (TK/TCL) scripting," *IOP Conference Series: Materials Science and Engineering*, vol. 42, 2018.

[78] 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*, vol. 32, no. 9, pp. 4387–4415, 2020.

[79] C. Chen, Y. Zhang, M. R. Khosravi, Q. Pei, and S. Wan, "An intelligent platooning algorithm for sustainable transportation systems in smart cities," *IEEE Sensors Journal*, 2020.

[80] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking:

past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[81] C. Jayapal, S. Jayavel, and V. P. Sumathi, "Enhanced service discovery protocol for MANET by effective cache management," *Wireless Personal Communications*, vol. 103, no. 2, pp. 1517–1533, 2018.

[82] T. T. Nguyen, V. N. Ha, L. B. Le, and R. Schober, "Joint data compression and computation offloading in hierarchical fog-cloud systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 293–309, 2020.

[83] T. Q. Thinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6652038, 12 pages
https://doi.org/10.1155/2021/6652038

Review Article

Advanced Power Management and Control for Hybrid Electric Vehicles: A Survey

Jielin Jiang[1,2] **Qinting Jiang**[1] **Jinhui Chen**[1] **Xiaotong Zhou**[1] **Shengkai Zhu**[1] **and Tianyu Chen**[1]

[1]*School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China*
[2]*Jiangsu Colloborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing, China*

Correspondence should be addressed to Qinting Jiang; q.jiang@nuist.edu.cn

Received 30 October 2020; Revised 13 December 2020; Accepted 24 December 2020; Published 7 January 2021

Academic Editor: Shaohua Wan

With the trend of low emissions and sustainable development, the demand for hybrid electric vehicles (HEVs) has increased rapidly. By combining a conventional internal combustion engine with one or more electric motors powered by a battery, HEVs have the advantages over traditional vehicles in better fuel economy and lower tailpipe emissions. Nevertheless, the power management strategies (PMSs) for conventional vehicles which mainly focus on the efficiency of internal combustion engine are no longer applicable due to the complex internal structure of HEVs. Hence, a large number of novel strategies appropriate for HEVs have been surveyed, but most of the researches concentrate on discussing the classifications of PMSs and comparing their cons and pros. This paper presents a comprehensive review of power management strategies adopted in HEVs aiming at specific challenges for the first time. The categories of the existing PMSs are presented based on the different algorithms, followed by a brief study of each type including the analysis of its pros and cons. Afterwards, the implementation and optimization of power management strategies aiming at proposed challenges are introduced in detail with the description of their optimization objectives and optimized results. Finally, future directions and open issues of PMSs in HEVs are discussed.

1. Introduction

Currently, due to the dramatic expansion of the world population and economic boom in most countries, the ownership of private cars is rising sharply, which imposes a serious burden on energy distribution and environmental protection [1]. To be specific, conventional cars are mainly powered by burning fossil fuels (diesel and petrol), but these fuels are reserved limited and nonrenewable [2]. Additionally, poisonous gases (e.g., CO, CnHm, and NOx) emitted by vehicles not only generate air pollution and pose enormous threats to human health but also exacerbate the greenhouse effect which increases the likelihood of global climate disasters (e.g., hurricane, tsunami, and rise of sea level).

Hybrid electric vehicles (HEVs) are considered as one of the most innovative solutions to the above challenges. Compared with the single power structure of traditional fuel-based vehicles, the power supply system of HEV is composed of several parts such as generator, internal combustion engine, and converter [3]. Such internal structure enriches the energy sources of HEV, enabling it to be driven by both electricity and heat, thus reducing the consumption of fossil fuels and emissions. Nevertheless, the power management strategies for conventional vehicles, which mainly focus on the efficiency of internal combustion engine, are no longer applicable due to the complex internal structure of HEV. New strategies are not only required to optimize the internal combustion engine but also to take the power of the battery, the flow, and the distribution of energy and collaboration of internal components (e.g., generator and internal combustion engine) into consideration [4]. Multiple management objectives greatly increase the complexity. The four main challenges in the HEV power management are as follows:

(1) Real-time optimization. Compared with static optimization, real-time optimization can adjust the

power management strategy according to driving conditions, thus significantly improving the timeliness of the strategy [5]. Nevertheless, limited by existing technologies (e.g., GPS and BIMS), it is impossible to accurately predict, analyze, and assess future driving conditions which include road conditions, traffic flow, and surrounding environment. Consequently, real-time optimization adaptive to dynamic driving conditions is challenging [6]

(2) Battery durability. Compared with the constant load conditions, the durability of fuel batteries tends to be significantly reduced under a dynamic loading condition [7]. Moreover, frequent charging and discharging process, switching voltages, and the flow of energy tend to accelerate battery aging

(3) Computation load. The computation load generated by some power management strategies, such as dynamic programming (DP), is likely to increase substantially with the expansion of optimal objectives [8]. Additionally, since current vehicular networks have limited computation capabilities, they may have difficulty in processing such large scale data instantly. Hence, most of these strategies are merely limited to theoretical analysis instead of practical operation [9]

(4) Multiple energy sources. Different from traditional fuel-based vehicles, HEV driven by multiple power sources has various energy flows and transitions internally. Hence, under diverse driving conditions, the cooperation of internal components (e.g., generator and internal combustion engine) and energy distribution tend to be more complicated [10]

A large number of power management strategies (PMSs) of HEV have been surveyed, but the current works are mainly focused on discussing the classifications of PMSs and comparing their cons and pros [11, 12]. No scholars have presented a comprehensive and thorough review of power management strategies aiming at specific challenges. To bridge this gap, a comprehensive and concrete survey of the recent research efforts on power management strategies in HEV in terms of above challenges is conducted in this paper, providing explicit research directions for later scholars [13].

The paper is organized as follows. Part II introduces the internal system framework of HEV, including power supply system, the generation of energy, and the cooperation of each part. Part III presents an overview of power management strategies of HEVs, providing their classifications and comparisons. After that, part IV reviews the implementation and optimization of power management strategies according to specific challenges raised above. Finally, part V discusses future trends and open issues of power management strategies in HEVs.

2. Power System Configuration

Due to the low dependency on fossil fuels [14], electric vehicles (EVs), especially hybrid electric vehicles, are recognized as alternatives for conventional vehicles. Hybrid electric vehicles can be commonly classified into three types: series hybrid electric vehicles (SHEVs), parallel hybrid electric vehicles (PHEVs), and series parallel hybrid electric vehicles (SPHEVs) [15].

2.1. Series Hybrid Electric Vehicles. The power system of a series hybrid electric vehicle consists of several batteries, a decoupled-from-wheel engine, an electric generator, and a motor [16]. The structure and the energy flow are shown in Figure 1. The main driving power for the wheels is directly provided by the battery pack rather than the engine. Additionally, the engines of SHEVs are utilized to drive the electric generators to charge the battery pack. The power released by the battery pack will then drive the motor to provide necessary energy and torque for the wheels [17].

During the energy conversion process, the energy forms are totally transformed three times, more than that of conventional vehicles. Due to such energy conversion mechanism, the engine is able to work smoothly. Thus, the redundant energy consumption caused by the external environment can be avoided to some extent. Nevertheless, if a SHEV runs at a high speed, this conversion process will reduce the energy efficiency, making it even lower than that of conventional vehicles. Based on the advantages and disadvantages analyzed above, this kind of HEVs can adapt to different situations when running at a low speed. For instance, the SHEVs technology tends to be utilized in the application of city-buses where frequent starts together with a low speed are demanded.

2.2. Parallel Hybrid Electric Vehicles. Different from SHEVs, PHEVs are driven by electric power and traditional heat energy simultaneously. Namely, that the energy sources of a PHEV are both the battery pack and the engine. The electricity stored in the battery pack is delivered to the motor. Meanwhile, the engine distributes its energy to the wheels and the electric generator to charge the batteries (Figure 2). Then, the power provided by the batteries and the engine works together to drive the wheels [18].

Owing to the design of multiply energy sources, the energy efficiency of PHEVs is higher than that of SHEVs. Nevertheless, the energy efficiency will be reduced if the state of charge (SOC) of the battery pack is low. Additionally, a battery pack with perfect SOC helps PHEVs work under a high-efficiency state. Hence, maintaining the SOC of the battery pack is another critical issue [19].

2.3. Series-Parallel Hybrid Electric Vehicles. The power system of a SPHEV is usually composed of an engine, two motors, two generators, and a battery pack. Such system lets the SPHEVs obtain the advantages of both SHEVs and PHEVs. SPHEVs are capable of working in different modes under various situations. Thus, SHEVs are more flexible in mode operation and more environmentally friendly [20]. The power supply of SPHEVs can be mechanical, electrical, or the both. For instance, when the battery pack is able to meet the energy requirements, the pure electric mode will be chosen; when the state of charge of the battery pack is

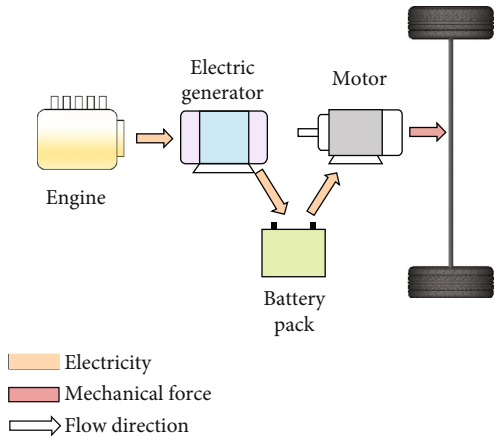


Figure 1: The power system diagram of SHEVs.

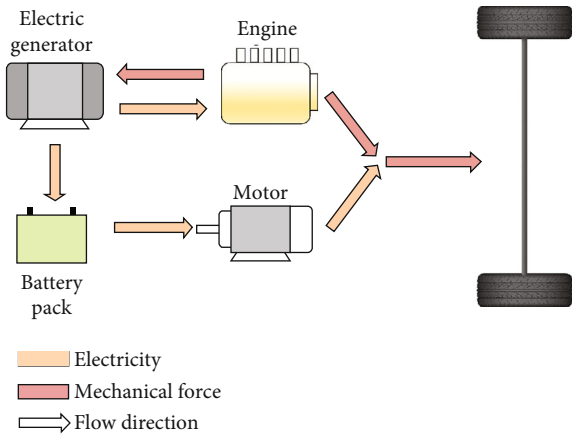


Figure 2: The power system diagram of PHEVs.

low or more power is needed, the engine will be turned on to satisfy the demand. However, flexible mode operation tends to cause the challenges of mode chosen and energy arrangement. Additionally, SPHEVs have some disadvantages both in production and economy. Firstly, the structure is of high cost, which may result in the difficulty of mass production. Hence, the widespread adoption will be a challenge. Moreover, the complexity of the structure leads to a technology monopoly between different automobile manufacturers, so the energy consumptions of vehicles from different companies may vary greatly.

The main structure of SPHEVs is shown in Figure 3. The transducer can provide the supply voltage needed by the motor owing to an internal inverter. Motors can be used as electric generators if necessary, so it is called as motor/generator (MG). The planetary gear train is a special structure which works as a power split device [21]. The rotation axis of the planetary gear carrier is connected to the engine and makes the planetary gears work together with the sun gear. The rotation axis of the sun gear is connected to MG1. The sun gear generates electric energy with the assistance of the engine. Meanwhile, the rotation axis of the planetary gear carrier is linked to MG2. The planetary gear provides power to the wheels.

As mentioned above, the complex structure may cause a series of problems. Firstly, the changeable driving condition requires that the driving mode should be chosen wisely by the control system. Moreover, the cooperation between electricity and conventional energy may not be smoothly enough, which often leads to low energy efficiency and more emissions.

3. Overview of HEV Power Management Strategies

3.1. Dynamic Programming. Dynamic programming (DP) is an optimization method commonly used in multistage decision-making process [22]. The optimization of energy consumption and emissions in HEV can just be regarded as such process in the discrete-time format [23]. Since DP is capable of finding the global optimum accurately, it is frequently applied in the power management of HEV to find the control solution. The state transition equation in HEV is showed in (1)

$$x(k+1) = f(x(k), u(k)), \tag{1}$$

where $u(k)$ is the vector of control variables, such as the desired output torque from the engine and the gear shift command to the transmission, and $x(k)$ is the state vector of the system. The optimization goal of DP is to find the control input $u(k)$ to minimize a cost function which consists of energy consumption and emissions. The cost function can be written as

$$J = \sum_{k=0}^{N-1} L(x(K), U(K)), \tag{2}$$

where N is the duration of the driving cycle, and L is the instantaneous cost including energy consumption and emissions which should be minimize. By leveraging the property that DP is based on Bellman's principle of optimality, we can easily obtain the optimal strategy. Specifically, we break down the original problem into several subproblems. The subproblem which involves only the last stage is solved primarily. Then, the subproblems involving the last two stages, the last three stages are gradually considered. The basic process of DP is presented in (3) and (4).

Step $N-1$:

$$J^*_{N-1}(x(N-1)) = \min_{u(N-1)} [L(x(N-1), u(N-1))], \tag{3}$$

Step k, for $0 \le k < N-1$

$$J^*_k(x(k)) = \min_{u(k)} [L(x(k), u(k)) + J^*_{k+1}(x(k+1))]. \tag{4}$$

During the optimization, it is necessary to take some inequality constraints into consideration to ensure safe and smooth operation of engine, motor, and battery such as the engine torque, the motor torque, the battery state of charge, and the engine speed [24].

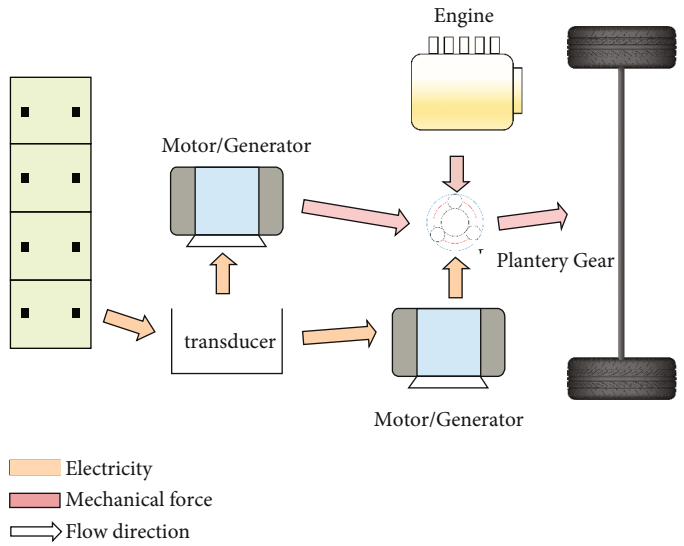


Figure 3: The power system diagram of SPHEVs.

Generally, DP is difficult to be directly used for this optimization process because of the high number of states in the model, i.e., curse of dimensionality [25, 26]. Therefore, the feasible methods based on DP always have been improved.

3.2. Genetic Algorithm. Genetic algorithm (GA) is a method which searches for the optimal solution by simulating the natural evolution process [27]. HEV is able to reduce energy consumption and emissions by leveraging the method of GA [28, 29].

Different parameter settings are regarded as different individuals in GA; then, the gene of each individual is coded by its own setting [30]. For example, an individual's genes can be expressed in binary numbers, such as genes = 00110100, which corresponds to a value within certain range of parameters.

The value of the fitness function is related to energy consumption and emissions which determined by genes, i.e., individuals with low energy consumption, and emissions genes are more likely to survive [31].

The processes of GA include initialization, selection, crossover, and mutation. Firstly, the genes of individual genes are coded randomly, i.e., individual genes are evenly distributed. Secondly, every individual is evaluated by its value of fitness function which determines its probability of survival. Third, the new individual is elected by its survival probability which can calculate by

$$p[i] = \frac{f_i}{\sum_{i=1}^{n} f_i}, \tag{5}$$

where f_i is the value of i^{th} individual of fitness function, $p[i]$ represents the probability that the i^{th} individual is selected, and n is the number of individuals. Finally, a new generation will be formed after necessary crossover and mutation. The iteration will keep going until the best individual fulfills the termination criteria, i.e., the energy consumption and emissions meet the requirements.

GA has a certain dependence on the selection of the initial population and can be improved by combining some heuristic algorithms. Nevertheless, the feedback information may fail to be used in time. Thus, the search speed of the algorithm is relatively slow, and training time is required to increase to obtain a more accurate solution [32].

3.3. Particle Swarm Optimization. Particle swarm optimization (PSO) is a random search algorithm based on group collaboration. PSO has fast convergence speed and does not need the strict condition that the optimization function has differentiability. Hence, PSO is usually leveraged to optimize the control strategy in different situations [33, 34].

PSO algorithm first initializes a group of random particles, and then the particles follow the current optimal particles in the solution space to find the optimal solution through iteration. The iteration formula is given in (6)

$$\begin{aligned} v_{ij}(t+1) &= w v_{ij}(t) + c_1 r_1 \left[p_{ij} - x_{ij}(t)\right] + c_2 r_2 \left[p_{gj} - x_{ij}(t)\right], x_{ij}(t+1) \\ &= x_{ij}(t) + v_{ij}(t+1), \mathrm{j} = 1, 2, \cdots d, \end{aligned} \tag{6}$$

where w is the inertia weight factor, c_1 and c_2 are learning factors, r_1 and r_2 are random numbers.

Similar to GA, PSO also regards variables to be optimized as particles and keeps getting closer to the needed result, but the information sharing mechanisms of the two methods are different [35]. The information flow is one-way in PSO, and the entire search update process follows the current optimal solution. Hence, all particles converge to the optimal solution at a fast speed. However, PSO is also more likely to fall into a local optimum situation [36].

3.4. Fuzzy Control. Fuzzy control has been introduced to the power management in HEVs. The aims of fuzzy control are

to achieve high efficient work and meet certain requirements by adjusting the current and the voltage. Meanwhile, it tries to avoid affecting the performance and efficiency of the entire system [37]. Compared with rule control, fuzzy control can output ratio according to different operating conditions. Additionally, fuzzy rules are easy to be adjusted and is robust to model errors and inaccurate measurements [38, 39].

In fuzzy control, the controller performs the basic steps of fuzzy logic. Primarily, the inputs are fuzzified into membership functions; then, the fuzzy outputs are computed by expertise-based rules. Ultimately, the outputs are fuzzified to proportional control signals [40, 41].

Although fuzzy control can optimize power management or emissions to a certain extent, it relies heavily on rules based on experience and fails to adjust according to the actual situation [22].

3.5. Equivalent Cost Minimization Strategy. Equivalent cost minimization strategy (ECMS) is a method frequently adopted in the PMSs of HEVs. By leveraging equivalent factors and predicting future costs to compensate the energy, ECMs convert the on-board electric energy depletion to an equivalent fuel consumption [42, 43].

The results revealed that the ability of ECMS can obtain a near optimal solution compared with DP at lower computational requirements. If powertrain components have constant efficiencies (mean value), then the cost to be minimized will be defined as (7)

$$C_{\text{tot}} = C_{\text{ICE}}(k(t), T_{th}(t)) + C_{\text{eq}}(k(t), T_{th}(t)) \tag{7}$$

where C_{ICE} is the real engine fuel consumption, and C_{eq} is electric motor equivalent fuel consumption. The design variables are the gear number $k(t)$, and the torque driver demands $T_{\text{th}}(t)$. The equivalence of electric energy is calculated by different charge or discharge processes of the battery

$$C_{\text{eq}}(k(t), T_{\text{th}}(t)) = \begin{cases} \dfrac{SFC_{\text{rech}}.P_e(\omega_e, T_e)}{\bar{\eta}_e\bar{\eta}_{\text{batt}} \cdot 3.6 \cdot 10^6} \forall T_e < 0 \\ \dfrac{SFC_{\text{dis}} \cdot P_e(\omega_e, T_e)\bar{\eta}_e\bar{\eta}_{\text{batt}}}{3.6 \cdot 10^6} \forall T_e \geq 0 \end{cases} \tag{8}$$

where SFC_{rech} and SFC_{dis} are the recharge and discharge mean specific fuel consumption, $\bar{\eta}_e$ and $\bar{\eta}_{\text{batt}}$ are the mean efficiency of electric motor and battery, and P_e is the motor power at torque T_e. The main challenge of ECMS is to consider the efficiency of each component and the dynamics of the power supply to estimate these equivalent factors.

3.6. Reinforcement Learning. Reinforcement learning (RL) is developed from theories of animal learning and parameter disturbance adaptive control. Its basic principles are as follows: if a certain behavior strategy of the agent leads to positive rewards (reinforcing signals) in the environment, then the tendency of the agent to produce such behavior strategy in the future will be strengthened. The goal of the agent is to find the optimal strategy in each discrete state to maximize the expected sum of discount rewards [44, 45].

The transition probability matrix can be expressed by

$$\begin{cases} p_{i,j} = \dfrac{M_{i,j}}{M_i} \\ M_i = \displaystyle\sum_{j=1}^{N} M_{i,j} \end{cases}. \tag{9}$$

The optimal value of states is defined as the expected sum of discount rewards which can be represented as (10)

$$V^*(s) = \min_{\pi} E\left(\sum_{t=t_0}^{t=t_f} \gamma^t r_t\right), \tag{10}$$

where π is a policy, and $\gamma \in [0, 1]$ is the discount factor.

The low sampling efficiency and requirements of huge learning time restrict the usage of RL. Therefore, RL needs to be leveraged under the right circumstances.

4. The Implementation and Optimization of PMSS

Although the HEVs have made great progress in improving fuel economy, reducing emissions, and achieving better vehicle performance, they still face significant challenges in power management. The challenges include real-time optimization, battery durability, computational load, and the power allocation among multienergy sources. To this endeavor, with the continuously advancing investigations, the novel power management strategies have been proposed. In this section, the advanced power management strategies aiming to address these challenges are discussed.

4.1. The Real-Time Optimization. In the real-time optimization, the optimal solution is often obtained based on the forecast of future conditions. It indicates that the future driving conditions, such as traffic conditions, road grade, and surrounding environment, are prerequisites for the real-time optimization. Accordingly, the methods utilized for the prediction of the future information are essential for the optimization of the power management strategy in real time.

Since the Markov chain is able to predict the power demand and vehicle velocity under stochastic circumstances, the strategies based on the Markov chain have been proposed. Zeng and Wang [19] proposed a stochastic model predictive strategy for the PHEV model under the hilly driving environment. The strategy modeled the grade, the speed change, and the stop and turning information as a Markov chain and applied a stochastic dynamic programming (SDP) strategy to maintain the battery SOC. Zou et al. [46] presented a three-dimensional Markov chain driving model for the tracked vehicles, where a nearest-neighborhood method was utilized to update an online transition probability matrix, and a SDP method for the tracked vehicles validated the reliability of the nearest-neighborhood approach. Additionally, Li et al. [47] designed a novel driving-

behavior-aware model predictive control method. The K-means was utilized to classify driving behaviors, and the Markov chain was employed to obtain driver models under different driving behaviors.

Due to the strong abilities in predicting and modeling, the artificial intelligence has been employed to forecast the driving cycles. Chen et al. [48] proposed a particle swarm optimization algorithm to optimize a rule-based power management strategy under a certain driving cycle. Meanwhile, a driving condition recognition algorithm was employed to identify real-time driving conditions by fuzzy logic. To address the problem that the thresholds are sensitive to the different driving cycles, a dynamic optimal parameter algorithm was established. Sun et al. [49] designed a velocity predictor based on a neural network to predict the short-term future driving behaviors. The velocity predictor was combined with adaptive-ECMS to provide temporary driving information for real-time equivalence factor adaptation. Liu et al. [50] presented a reinforcement learning-based adaptive energy management for a hybrid electric, where fuel consumption was minimized over different driving schedules to guarantee power demand. Table 1 lists the main strategies designed for the real-time optimization.

4.2. The Battery Durability. The HEV system requires high energy capacities for long driving distances and high power capacities for accelerating, climbing, or braking. These requirements (high energy capacities and high power capacities) keep the battery in frequent discharge-charge condition. Nevertheless, the battery durability is impaired by the high discharge-charge rates, leading to a reduction in fuel economy. Hence, a proper power management for HEV is required to fulfill the durability of the battery.

To deal with the battery durability, a system-level design is essential. Capasso et al. [51] developed an optimal control strategy, which exploits the off-line solution of an isoperimetric problem and dynamically optimizes the battery durability via reducing peak current. The results showed the effectiveness of the strategy in reducing the high charging/-discharging current peaks to increase the battery durability. In addition, Zhang et al. [52] proposed a hysteresis control strategy for HEV with three fuel cell stacks, where each fuel cell stack works at a fixed operating point, and its active time is shortened by on-off switching control. Combined with the power capability and SOC states, Wang et al. [53] designed a finite state machine-based management strategy and presented an optimal oxygen excess ratio control to maximize the net power of fuel cell. The simulated results indicated that the method guarantees the required power during the driving cycles. Additionally, by taking the battery durability into consideration, the power management strategies can obtain the near-optimal solution. Zhang et al. [54] proposed an optimal power management strategy based on the DP algorithm and verified by the different battery SOC and battery state-of-health (SOH) conditions, which guarantees a better strategy control performance. To optimize the control of the fuel cell system, Robin et al. [55] designed a mechanistic catalyst dissolution model to predict the lifetime of fuel cell and utilized a model inversion to forecast the performance loss. The mechanistic catalyst dissolution model successfully passed the battery durability test in dynamic operation conditions. The effect and strategies that have been put forward in developing the battery durability are shown in Table 2.

4.3. Computational Load. Due to the curse of dimensions, the data required to support a reliable result tends to multiply exponentially with the increase in variables. A mass of data leads to a high computation load or even an inability to calculate. Consequently, some power management strategies (e.g., dynamic programming) fail to be applied for the power management of HEV without the optimization for computational complexity in practice.

Larsson et al. [56] put forward a method based on local approximation of the gridded cost-to-go and utilized local approximations at the appropriate control signal to reduce the quantized interpolation. Combined with the particle swarm optimization (PSO), Yang et al. [57] proposed a rapid-DP optimization strategy to select the optimal control state of the motor and further improved fuel economy of the vehicle.

With the proposal of level-set DP in [58], the computing time of DP was decreased by 300 times. Nevertheless, more challenges, e.g., the Markov and standardization problems, were raised. To deal with these problems, Zhou et al. [59] proposed a unified solution method, where the Markov characteristics of DP were utilized to construct a unified equation of state. A massive amount of data in the computing was reduced by filters based on state variables and control variables. This method was faster than the conventional DP, reducing computation time by 96.48% and 23.44% compared with basic DP and level-set DP.

Due to the low computational complexity of neural network, Li et al. [60] presented a power management strategy based on reinforcement learning without worrying curse of dimensionality in complex environments, where stochastic gradient descent and experience replay were adopted to guarantee the accuracy and stability of the method.

The strategies to address the computational load discussed are listed in Table 3.

4.4. The Power Allocation of Multiple Energy Sources. Compared with EVs, PHEVs have longer driving ranges. On the one hand, the engine allows the vehicle to work when the battery SOC is at a low state, which is similar to the situation of conventional vehicles. On the other hand, the engine will be turned off, and the vehicle will be driven by the electric power system when the speed or the power demand is low. Therefore, the driving performance depends on the power allocation among multienergy sources.

Many valuable works related to power management for HEVs with multienergy sources, where intelligent strategies (e.g., fuzzy logic, dynamic programming, and particle swarm optimization [48]) are utilized to optimize energy allocation among multiple sources, have been widely conducted. Nevertheless, most of the works investigate power management strategies for the HEVs powered by the battery and engine or the battery and ultracapacitor, and few of them aim at

Table 1: Main strategies proposed to deal with the real-time optimization.

Reference	Solution	Highlights
[19]	Markov chains and stochastic dynamic programming	(1) Model the road grade as a Markov chain (2) Maintain the SOC within its boundary
[46]	Markov chains and nearest-neighborhood method	(1) Present a three-dimensional Markov chain driver model (2) Update transition probability matrix online
[47]	Markov chains	(1) Classify eight typical driving behaviors (2) Establish driver models under different driving behaviors
[48]	Particle swarm optimization and fuzzy logic	(1) Pay attention to uncertain driving condition (2) Avoid thresholds sensitive to driving cycles
[49]	Neural network	(1) Design a velocity predictor (2) Real-time adaptation
[50]	Reinforcement learning	(1) Present a control-oriented dynamic model (2) Method adaptability under different driving conditions

Table 2: Main strategies proposed to deal with the battery durability.

Reference	Solution	Highlights
[51]	Isoperimetric optimization	(1) Dynamically optimize battery durability (2) Reduce high discharge-charge current peaks
[52]	Optimal control	(1) Propose a hysteresis power management and control strategy (2) Provide a novel configuration
[53]	Finite state machine	(1) Consider power capability and SOC (2) Maximize the net power
[54]	Dynamic programming	(1) Consider battery durability (2) Verify strategies under SOC and SOH condition
[55]	Lifetime prediction	(1) Design a mechanistic catalyst dissolution model (2) Forecast the performance loss

Table 3: Main strategies proposed to deal with deal with the computational load.

Reference	Solution	Highlights
[56]	Local approximation	(1) Reduce interpolation (2) Reduce computing time by two orders of magnitude
[57]	Rapid-DP and particle swarm optimization	(1) Propose a joint optimization (2) Improve fuel economy
[58]	Level-set function	(1) Decrease computing time of DP by 300 times
[59]	A unified DP model	(1) Solve the Markov problem in DP (2) Reduce computation data
[60]	Reinforcement learning	(1) Avoid the curse of dimensionality (2) Complex environment stability

the power management strategies of vehicles with more than two sources.

In the hybrid energy storage system (HESS), a battery and an ultracapacitor are combined to reduce the charge rate of the battery. To deal with the energy allocation problem among the HESS and engine-generator, Zhang and Xiong [61] proposed a hierarchical control strategy, where a fuzzy logic controller was employed for classifying the driving patterns, and the DP method was utilized to develop control strategies for different driving blocks. However, in [61], the HESS is viewed as a single source, and the power of the battery and ultracapacitor was determined by the deterministic required power. Thus, an integrated power management strategy where the battery and the ultracapacitor were regarded as difference power sources [62] was designed, including HESS and an assistance power unit (APU). Utilizing a model predictive control (MOC) controller, the power allocation between battery and ultracapacitor could be realized, while the output power of HESS and APU is allocated by the rule-based strategy. To obtain the real-time power allocation between the battery and the ultracapacitor for the HESS, Xiong et al. [63] presented a reinforcement learning-based energy management, which could learn current driving power information and update the strategy in time.

Table 4: Main strategies proposed to deal with deal with the multiple energy sources.

Reference	Solution	Highlights
[61]	Driving pattern recognition and dynamic programming	(1) Propose a hierarchical control strategy for HESS (2) Classify different driving patterns
[62]	Model predictive control and dynamic programming	(1) Design an assistance power unit (2) Pay attention to the power allocation between battery and ultracapacitor (3) Present an MPC controller
[63]	Reinforcement learn	(1) Real-time power allocation strategy
[64]	Fuzzy logic control genetic algorithm	(1) Propose a novel method for FCS
[43]	Equivalent consumption minimization strategy	(1) Pay attention to the PEMFC (2) Improve overall efficiency

Ahmadi et al. [64] proposed a novel power allocation method for the fuel-cell vehicle powered by the fuel-cell system (FCS), battery, and ultracapacitor, and implemented an intelligent control technique based on fuzzy logic control, which determines the required power for FCS and ultracapacitor. Additionally, since the proton exchange membrane fuel cell (PEMFC) plays an important role in developing the fuel-cell vehicle, Li et al. [43] designed a system, where PEMFC, two batteries, and two supercapacitors were combined to avoid the rapid changes of power demand, and ECMS was utilized to achieve better energy efficiency of the overall system.

The strategies are summarized and shown in Table 4.

5. Open Issues and Challenges

This section puts forwards some remaining challenges of power management strategies in HEVs that should be taken into account, and the open issues and future trends will also be discussed.

5.1. System Stability. The power system in HEVs consists of multienergy sources (e.g., motor, engine, and battery), power switching unit, and converters. These components tend to be affected by the changes in parameters such as temperature, discharge-charge rate, and load variation [10]. Consequently, the disturbance from the parameters leads to changing power demands, battery overload, and low power quality, which eventually results in the instability of the vehicle power system [65]. To deal with the challenge for system stability, the design of the power management strategies is ought to take the terminal cost and stability constraints into account to guarantee the stable system operation [66, 67].

5.2. System Robustness. The robustness of the power management strategies refers to the ability of the control system to maintain the model stability and resist system noises and disturbances [68]. Nevertheless, most of the power management and control strategies for the HEVs are simulated in the specific scenarios, ignoring the uncertainties that may occur in a real scenario. Such static power management strategies are likely to cause a poor vehicle performance.

On the one hand, some power management strategies, such as Markov chains, are based on the collected data. Therefore, if the real driving conditions differ from the collected data of the driving cycles, the algorithms fail to the optimization [69]. On the other hand, the components in the system configuration, such as generators, batteries, and capacitors, always age and wear out during the operation, resulting in uncertainties of the configuration and parament [70, 71].

Accordingly, based on the power management strategy for HEVs, the robust control strategy adopted to the real scenario requires investigating.

5.3. Edge Computing. Currently, technologies related to edge computing are being studied extensively. Different from the centralized computing model of cloud computing, mobile users look for nearby available devices and base stations to offload the current computation tasks in an edge computing scenario [72]. Data transfer overhead and latency are greatly reduced as the edge nodes are closer to the user [73].

Due to the increasing requirements of the low latency and computing for power management strategies in HEV, it is necessary to move the computational nodes from the cloud data centers to the edge nodes [74, 75]. Additionally, computing offloads on edge devices enhance the responsiveness of the service while significantly reducing the energy loss caused by data transfer. Meanwhile, the distributed computing nodes have the potential to enable the robustness of the power management strategies to guarantee the vehicle safety, real-time optimization, and fuel economy [76].

5.4. Smart Grid. With the rapid expansion of strategic emerging industries like hybrid electric vehicles, great importance has been attached to an electric automation level in enhancing overall efficiency and improving electricity supply reliability [77].

Smart grid is a modern power grid featured as being automatic, interactive, and IT-based. It is composed of different types of generation sources along with introduction of information and communication technologies (ICT). Supported by the intelligent control and IT platform, smart grid involves six segments including power generation, transformation, transmission, distribution, dispatching, and consumption. In the scenario of smart grid, the charging efficiency of hybrid electric vehicles will be significantly improved, followed by the reduced charging cost [78].

5.5. Battery Aging. Battery aging is a common issue in many types of batteries. During the charging and discharging process, chemical reactions take place inside the battery constantly which corrodes the cathode of the battery until the cathode completely deteriorates. Batteries should be replaced regularly if the aging issue is serious. Thus, battery exerts an important influence on the overall cost of HEVs. Although many studies have been conducted on power management strategies of HEV, only a few of them take this issue into consideration [79].

One promising solution is to combine a supercapacitor with the battery. Compared to supercapacitor, battery has better energy density but poor power density to release energy sharply. Moreover, the cycling life of battery is much shorter. On the other hand, although supercapacitor has lower energy density, it generally has much higher power density. The combination of the both can play a complementary role.

6. Conclusion

The characteristics of low energy consumption and limited emissions of HEV make it a promising industry. Substituting HVEs for conventional fuel-based vehicles is expected to alleviate the current energy shortage and serious environmental pollution. The goal of this paper is to comprehensively study the power management strategies for HEVs aiming at specific challenges. The main challenges in power management strategies for HEVs are listed at the beginning. After a brief introduction on internal dynamic structures of HEV and an overview of existing power management strategies, the comparisons and experimental results of each method are also presented. Eventually, several open issues and future trends of HEVs are discussed.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant no. 61601235, in part by the Natural Science Foundation of Jiangsu Province of China under Grant no. BK20160972. Meanwhile, the authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RG-1441-331.

References

[1] R. Dominguez, J. Solano, and A. Jacome, "Sizing of fuel cell - ultracapacitors hybrid electric vehicles based on the energy management strategy," in *2018 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–5, Chicago, IL, 2018.

[2] A. Rezaei, J. B. Burl, M. Rezaei, and B. Zhou, "Catch energy saving opportunity in charge-depletion mode, a real-time controller for plug-in hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11234–11237, 2018.

[3] J. Guan and B. Chen, "Adaptive power management strategy based on equivalent fuel consumption minimization strategy for a mild hybrid electric vehicle," in *2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–4, Hanoi, Vietnam, 2019.

[4] S. S. George and M. O. Badawy, "A modular multi-level converter for energy management of hybrid storage system in electric vehicles," in *2018 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 336–341, Long Beach, CA, 2018.

[5] R. Ghaderi, M. Kandidayeni, M. Soleymani, and L. Boulon, "Investigation of the battery degradation impact on the energy management of a fuel cell hybrid electric vehicle," in *2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–6, Hanoi, Vietnam, 2019.

[6] D. He, Y. Zou, J. Wu, X. Zhang, Z. Zhang, and R. Wang, "Deep Q-learning based energy management strategy for a series hybrid electric tracked vehicle and its adaptability validation," in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–6, Detroit, MI, USA, 2019.

[7] J. Guo, H. He, and C. Sun, "ARIMA-based road gradient and vehicle velocity prediction for hybrid electric vehicle energy management," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5309–5320, 2019.

[8] J. Oncken and B. Chen, "Real-time model predictive powertrain control for a connected plug-in hybrid electric vehicle," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8420–8432, 2020.

[9] J. Wu, J. Ruan, N. Zhang, and P. D. Walker, "An optimized real-time energy management strategy for the power-split hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1194–1202, 2019.

[10] S. Zhou, Z. Chen, D. Huang, and T. Lin, "Model prediction and rule based energy management strategy for a plug-in hybrid electric vehicle with hybrid energy storage system," *IEEE Transactions on Power Electronics*, 2020.

[11] H. H. Nguyen, J. Kim, G. Hwang, S. Lee, and M. Kim, "Research on novel concept of hybrid electric vehicle using removable engine-generator," in *2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–5, Hanoi, Vietnam, 2019.

[12] Q. Sun, J. Wu, C. Gan, J. Si, J. Guo, and Y. Hu, "Cascaded multiport converter for SRM-based hybrid electrical vehicle applications," *IEEE Transactions on Power Electronics*, vol. 34, no. 12, pp. 11940–11951, 2019.

[13] L. Zhang, X. Hu, Z. Wang, F. Sun, J. Deng, and D. G. Dorrell, "Multi-objective optimal sizing of hybrid energy storage system for electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1027–1035, 2018.

[14] P. G. Anselma, Y. Huo, J. Roeleveld, G. Belingardi, and A. Emadi, "Integration of on-line control in optimal design of multimode power-split hybrid electric vehicle powertrains," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3436–3445, 2019.

[15] A. Macias Fernandez, M. Kandidayeni, L. Boulon, and H. Chaoui, "An adaptive state machine based energy management strategy for a multi-stack fuel cell hybrid electric vehicle,"

IEEE Transactions on Vehicular Technology, vol. 69, no. 1, pp. 220–234, 2020.

[16] J. Gissing, P. Themann, S. Baltzer, T. Lichius, and L. Eckstein, "Optimal control of series plug-in hybrid electric vehicles considering the cabin heat demand," *IEEE Transactions on Control Systems and Technology*, vol. 24, no. 3, pp. 1126–1133, 2016.

[17] Y. Li, H. He, J. Peng, and H. Wang, "Deep reinforcement learning-based energy management for a series hybrid electric vehicle enabled by history cumulative trip information," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7416–7430, 2019.

[18] L. Li, C. Yang, Y. Zhang, L. Zhang, and J. Song, "Correctional DP-based energy management strategy of plug-in hybrid electric bus for city-bus route," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 7, pp. 2792–2803, 2015.

[19] X. Zeng and J. Wang, "A parallel hybrid electric vehicle energy management strategy using stochastic model predictive control with road grade preview," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2416–2423, 2015.

[20] H. Zhang, Y. Zhang, and C. Yin, "Hardware-in-the-loop simulation of robust mode transition control for a series–parallel hybrid electric vehicle," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1059–1069, 2016.

[21] A. Gayebloo and A. Randan, "Superiority of dual-mechanical-port-machine-based structure for series–parallel hybrid electric vehicle applications," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 589–602, 2016.

[22] M. Passalacqua, D. Lanzarotto, M. Repetto, L. Vaccaro, A. Bonfiglio, and M. Marchesoni, "Fuel economy and EMS for a series hybrid vehicle based on supercapacitor storage," *IEEE Transactions on Power Electronics*, vol. 34, no. 10, pp. 9966–9977, 2019.

[23] D. S. Mendoza, P. Acevedo, J. S. Jaimes, and J. Solano, "Energy management of a dual-mode locomotive based on the energy sources characteristics," in *2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–4, Hanoi, Vietnam, 2019.

[24] Z. H. C. Daud, Z. Asus, S. A. A. Bakar, N. A. Husain, I. I. Mazali, and D. Chrenko, "Thermal characteristics of a lithium-ion battery used in a hybrid electric vehicle under various driving cycles," *IET Electrical Systems in Transportation*, vol. 10, no. 3, pp. 243–248, 2020.

[25] A. Rezaei, J. B. Burl, B. Zhou, and M. Rezaei, "A new real-time optimal energy management strategy for parallel hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 830–837, 2019.

[26] C. Zhu, F. Lu, H. Zhang, J. Sun, and C. C. Mi, "A real-time battery thermal management strategy for connected and automated hybrid electric vehicles (CAHEVs) based on iterative dynamic programming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8077–8084, 2018.

[27] R. Liessner, A. Lorenz, J. Schmitt, A. M. Dietermann, and B. Baker, "Simultaneous electric powertrain hardware and energy management optimization of a hybrid electric vehicle using deep reinforcement learning and Bayesian optimization," in *2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–6, Hanoi, Vietnam, 2019.

[28] S. Yang, M. Li, B. Xu, B. Guo, and C. Zhu, "Optimization of fuzzy controller based on genetic algorithm," in *2010 International Conference on Intelligent System Design and Engineering Application*, pp. 21–28, Changsha, 2010.

[29] Y. Cheng, C. Lai, and J. Teh, "Optimization of control strategy for hybrid electric vehicles based on improved genetic algorithm," in *2017 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–4, Belfort, 2017.

[30] M. Yue, S. Jemei, and N. Zerhouni, "Health-conscious energy management for fuel cell hybrid electric vehicles based on prognostics-enabled decision-making," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11483–11491, 2019.

[31] S. Uebel, N. Murgovski, B. Bäker, and J. Sjöberg, "A two-level MPC for energy management including velocity control of hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5494–5505, 2019.

[32] S. Nazari, J. Siegel, and A. Stefanopoulou, "Optimal energy management for a mild hybrid vehicle with electric and hybrid engine boosting systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3386–3399, 2019.

[33] G. Celli, E. Ghiani, F. Pilo, G. Pisano, and G. G. Soma, "Particle swarm optimization for minimizing the burden of electric vehicles in active distribution networks," in *2012 IEEE Power and Energy Society General Meeting*, pp. 1–7, San Diego, CA, 2012.

[34] M. Zhou, H. Zhang, and X. Wang, "Research on fuzzy energy management strategy of parallel hybrid electric vehicle," in *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, pp. 967–971, Harbin, 2011.

[35] J. Qi, C. Lai, B. Xu, Y. Sun, and K. Leung, "Collaborative energy management optimization toward a green energy local area network," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5410–5418, 2018.

[36] S. Xie, S. Qi, and K. Lang, "A data-driven power management strategy for plug-in hybrid electric vehicles including optimal battery depth of discharging," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3387–3396, 2020.

[37] A. A. Ferreira, J. A. Pomilio, G. Spiazzi, and L. de Araujo Silva, "Energy management fuzzy logic supervisory for electric vehicle power supplies system," *IEEE Transactions on Power Electronics*, vol. 23, no. 1, pp. 107–115, 2008.

[38] W. Lee, H. Jeoung, D. Park, and N. Kim, "An adaptive concept of PMP-based control for saving operating costs of extended-range electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11505–11512, 2019.

[39] Q. Zhang and G. Li, "Experimental study on a semi-active battery-supercapacitor hybrid energy storage system for electric vehicle application," *IEEE Transactions on Power Electronics*, vol. 35, no. 1, pp. 1014–1021, 2020.

[40] H. N. de Melo, J. P. F. Trovão, P. G. Pereirinha, H. M. Jorge, and C. H. Antunes, "A controllable bidirectional battery charger for electric vehicles with vehicle-to-grid capability," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 114–123, 2018.

[41] J. Chen, C. Xu, C. Wu, and W. Xu, "Adaptive fuzzy logic control of fuel-cell-battery hybrid systems for electric vehicles," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 292–300, 2018.

[42] M. A. Ali and D. Soffker, "Towards optimal power management of hybrid electric vehicles in real-time: a review on methods, challenges, and stateof-the-art solutions," *Energies*, vol. 11, no. 3, pp. 1–24, 2018.

[43] Q. Li, T. Wang, C. Dai, W. Chen, and L. Ma, "Power management strategy based on adaptive droop control for a fuel cell-

battery-Supercapacitor hybrid tramway," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5658–5670, 2018.

[44] P. G. Anselma, Y. Huo, J. Roeleveld, G. Belingardi, and A. Emadi, "Slope-weighted energy-based rapid control analysis for hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4458–4466, 2019.

[45] T. Liu, X. Tang, H. Wang, H. Yu, and X. Hu, "Adaptive hierarchical energy Management Design for a Plug-in Hybrid Electric Vehicle," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11513–11522, 2019.

[46] Y. Zou, Z. Kong, T. Liu, and D. Liu, "A real-time Markov chain driver model for tracked vehicles and its validation: its adaptability via stochastic dynamic programming," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 3571–3582, 2017.

[47] L. Li, S. You, C. Yang, B. Yan, J. Song, and Z. Chen, "Driving-behavior-aware stochastic model predictive control for plug-in hybrid electric buses," *Applied Energy*, vol. 162, pp. 868–879, 2016.

[48] Z. Chen, R. Xiong, and J. Cao, "Particle swarm optimization-based optimal power management of plug-in hybrid electric vehicles considering uncertain driving conditions," *Energy*, vol. 96, pp. 197–208, 2016.

[49] C. Sun, F. Sun, and H. He, "Investigating adaptive-ecms with velocity forecast ability for hybrid electric vehicles," *Applied Energy*, vol. 185, pp. 1644–1653, 2017.

[50] T. Liu, Y. Zou, D. Liu, and F. Sun, "Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7837–7846, 2015.

[51] C. Capasso, D. Lauria, and O. Veneri, "Optimal control strategy of ultra-capacitors in hybrid energy storage system for electric vehicles," *Energy Procedia*, vol. 142, pp. 1914–1919, 2017.

[52] H. Zhang, X. Li, X. Liu, and J. Yan, "Enhancing fuel cell durability for fuel cell plug-in hybrid electric vehicles through strategic power management," *Applied Energy*, vol. 241, pp. 483–490, 2019.

[53] Y. Wang, Z. Sun, and Z. Chen, "Energy management strategy for bat-tery/supercapacitor/fuel cell hybrid source vehicles based on finite state machine," *Applied Energy*, vol. 254, 2019.

[54] S. Zhang, R. Xiong, and J. Cao, "Battery durability and longevity based power management for plug-in hybrid electric vehicle with hybrid energy storage system," *Applied Energy*, vol. 179, pp. 316–328, 2016.

[55] C. Robin, M. Gerard, M. Quinaud, J. Darbigny, and Y. Bultel, "Proton exchange membrane fuel cell model for aging predictions: simulated equivalent active surface area loss and comparisons with durability tests," *Journal of Power Sources*, vol. 326, pp. 417–427, 2016.

[56] V. Larsson, L. Johannesson, and B. Egardt, "Analytic solutions to the dynamic programming subproblem in hybrid vehicle energy management," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1458–1467, 2015.

[57] Y. Yang, H. Pei, X. Hu, Y. Liu, C. Hou, and D. Cao, "Fuel economy optimization of power split hybrid vehicles: a rapid dynamic programming approach," *Energy*, vol. 166, pp. 929–938, 2019.

[58] P. Elbert, S. Ebbesen, and L. Guzzella, "Implementation of dynamic programming for n-dimensional optimal control problems with final state constraints," *IEEE Transactions on Control Systems and Technology*, vol. 21, no. 3, pp. 924–931, 2013.

[59] W. Zhou, L. Yang, Y. Cai, and T. Ying, "Dynamic programming for new energy vehicles based on their work modes part i: electric vehicles and hybrid electric vehicles," *Journal of Power Sources*, vol. 406, pp. 151–166, 2018.

[60] Y. Li, H. He, J. Peng, and H. Zhang, "Power management for a plug-in hybrid electric vehicle based on reinforcement learning with continuous state and action spaces," *Energy Procedia*, vol. 142, pp. 2270–2275, 2017.

[61] S. Zhang and R. Xiong, "Adaptive energy management of a plug-in hybrid electric vehicle based on driving pattern recognition and dynamic programming," *Applied Energy*, vol. 155, pp. 68–78, 2015.

[62] S. Zhang, R. Xiong, and F. Sun, "Model predictive control for power management in a plug-in hybrid electric vehicle with a hybrid energy storage system," *Applied Energy*, vol. 185, pp. 1654–1662, 2017.

[63] R. Xiong, J. Cao, and Q. Yu, "Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle," *Applied Energy*, vol. 211, pp. 538–548, 2018.

[64] S. Ahmadi, S. M. T. Bathaee, and A. H. Hosseinpour, "Improving fuel economy and performance of a fuel-cell hybrid electric vehicle (fuel-cell, battery, and ultra-capacitor) using optimized energy management strategy," *Energy Conversion and Management*, vol. 160, pp. 74–84, 2018.

[65] Q. Zhang, W. Deng, and G. Li, "Stochastic control of predictive power management for battery/supercapacitor hybrid energy storage systems of electric vehicles," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3023–3030, 2018.

[66] Z. Zhang, C. Guan, and Z. Liu, "Real-time optimization energy management strategy for fuel cell hybrid ships considering power sources degradation," *IEEE Access*, vol. 8, pp. 87046–87059, 2020.

[67] O. Salari, K. H. Zaad, A. Bakhshai, and P. Jain, "Filter design for energy management control of hybrid energy storage systems in electric vehicles," in *2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pp. 1–7, Charlotte, NC, 2018.

[68] J. J. Mwambeleko and T. Kulworawanichpong, "Battery and accelerating-catenary hybrid system for light rail vehicles and trams," in *2017 International Electrical Engineering Congress (iEECON)*, pp. 1–4, Pattaya, 2017.

[69] S. R. Marjani, M. Gheibi, V. Talavat, and M. Farsadi, "A novel hybrid intelligent method for static var compensator placement in distribution network with plug-in hybrid electrical vehicles parking," in *2015 Intl Aegean Conference on Electrical Machines Power Electronics (ACEMP), 2015 Intl Conference on Optimization of Electrical Electronic Equipment (OPTIM) 2015 Intl Symposium on Advanced Electromechanical Motion Systems (ELECTROMOTION)*, pp. 323–330, Side, Turkey, 2015.

[70] M. A. Saeed, N. Ahmed, M. Hussain, and A. Jafar, "A comparative study of controllers for optimal speed control of hybrid electric vehicle," in *2016 International Conference on Intelligent Systems Engineering (ICISE)*, pp. 1–4, Islamabad, 2016.

[71] J. Solano, D. Hissel, and M. Pera, "Energy management of an hybrid electric vehicle in degraded operation," in *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–4, Coimbra, 2014.

[72] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A distributed canny edge detector: algorithm and FPGA

implementation," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2944–2960, 2014.

[73] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The extended cloud: review and analysis of mobile edge computing and fog from a security and resilience perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586–2595, 2017.

[74] T. Bahreini, M. Brocanelli, and D. Grosu, "Energy-aware resource management in vehicular edge computing systems," in *2020 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 49–58, Sydney, Australia, 2020.

[75] Y. Cao, H. Song, O. Kaiwartya et al., "Mobile edge computing for big-data-enabled electric vehicle charging," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 150–156, 2018.

[76] D. Ahmad, S. Z. Hassan, A. Zahoor et al., "A bidirectional wireless power transfer for electric vehicle charging in V2G system," in *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–6, Swat, Pakistan, 2019.

[77] Z. Lv, H. Song, P. Basanta-Val, A. Steed, and M. Jo, "Next-generation big data analytics: state of the art, challenges, and future research topics," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1891–1899, 2017.

[78] X. Xu, B. Shen, S. Ding et al., "Service offloading with deep Q-network for digital twinning empowered Internet of Vehicles in edge computing," *IEEE Transactions on Industrial Informatics*, 2020.

[79] C. Chen, Y. Zhang, M. R. Khosravi, Q. Pei, and S. Wan, "An intelligent platooning algorithm for sustainable transportation systems in smart cities," *IEEE Sensors Journal*, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9858963, 1 page
https://doi.org/10.1155/2023/9858963

Retraction

Retracted: Improved Multiview Decomposition for Single-Image High-Resolution 3D Object Reconstruction

Wireless Communications and Mobile Computing

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Peng, K. Fu, Q. Wei, Y. Qin, and Q. He, "Improved Multiview Decomposition for Single-Image High-Resolution 3D Object Reconstruction," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8871082, 14 pages, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8871082, 14 pages
https://doi.org/10.1155/2020/8871082

Research Article

Improved Multiview Decomposition for Single-Image High-Resolution 3D Object Reconstruction

Jiansheng Peng, Kui Fu, Qingjin Wei, Yong Qin, and Qiwen He

School of Physics and Mechanical and Electronic Engineering, Hechi University, Yizhou 546300, China

Correspondence should be addressed to Jiansheng Peng; 1692759628@qq.com

Received 3 September 2020; Revised 3 December 2020; Accepted 13 December 2020; Published 28 December 2020

Academic Editor: Shaohua Wan

As a representative technology of artificial intelligence, 3D reconstruction based on deep learning can be integrated into the edge computing framework to form an intelligent edge and then realize the intelligent processing of the edge. Recently, high-resolution representation of 3D objects using multiview decomposition (MVD) architecture is a fast reconstruction method for generating objects with realistic details from a single RGB image. The results of high-resolution 3D object reconstruction are related to two aspects. On the one hand, a low-resolution reconstruction network represents a good 3D object from a single RGB image. On the other hand, a high-resolution reconstruction network maximizes fine low-resolution 3D objects. To improve these two aspects and further enhance the high-resolution reconstruction capabilities of the 3D object generation network, we study and improve the low-resolution 3D generation network and the depth map superresolution network. Eventually, we get an improved multiview decomposition (IMVD) network. First, we use a 2D image encoder with multifeature fusion (MFF) to enhance the feature extraction capability of the model. Second, a 3D decoder using an effective subpixel convolutional neural network (3D ESPCN) improves the decoding speed in the decoding stage. Moreover, we design a multiresidual dense block (MRDB) to optimize the depth map superresolution network, which allows the model to capture more object details and reduce the model parameters by approximately 25% when the number of network layers is doubled. The experimental results show that the proposed IMVD is better than the original MVD in the 3D object superresolution experiment and the high-resolution 3D reconstruction experiment of a single image.

1. Introduction

The three-dimensional reconstruction of a single image is a hotspot and a difficult point in the field of computer vision. The purpose of the three-dimensional reconstruction of a single image is to reconstruct the corresponding 3D model structure from a single RGB image or a single depth image. The early 3D reconstruction of objects used the multiview geometry (MVG) method, which mainly studied structure-from-motion (SFM) [1, 2] recovery and simultaneous localization and mapping (SLAM) [3]. In addition, 3D object reconstruction also has methods based on prior knowledge [4, 5]. These traditional methods are often limited to a certain class of object in the 3D reconstruction of a single image, or it is difficult to generate a 3D object with better precision. With the continuous development of deep learning technology, the technology has been widely used in recent years [6–14], such as video analysis [8], image processing [9–11], medical diagnosis and service [12, 13], and target recognition [14]. Applying these to actual scenarios will encounter problems of large energy consumption and long response time. Using edge computing can effectively solve these problems. In the era of big data, data generated at the edge (e.g., images) also requires artificial intelligence technology to release its potential. Some research attempts to combine edge computing and deep learning include intelligent video surveillance [15], food recognition systems, [16], and self-driving cars [17]. At present, most of the research on edge computing and deep learning focuses on object recognition in two-dimensional space. However, for applications such as self-driving and

virtual reality, 3D reconstruction is the core technology. In the 3D reconstruction of objects, many methods try to extend the convolution operation in the two-dimensional space to the three-dimensional space to generate 3D shapes [18–20] and have achieved good research results. These methods all use a convolution operation based on dense voxels. As the running time and memory consumed increase cubically with the improvement of voxel resolution, the resolution of the generated models is limited to $64 \times 64 \times 64$. In order to solve the problem that the model generated by this method is limited to low resolution, some studies have proposed a sparse 3D reconstruction method using octrees [21–23]. Recently, the generative adversarial network (GAN) has shown great potential in image generation, and Yu et al. [24] also extended it to the 3D reconstruction of a single image. For the 3D reconstruction of a single image using GAN, this method consumes huge computing resources and also has a long training time. At present, the application of edge computing [25, 26] may be a feasible solution to this problem. Applying edge computing to traditional 3D reconstruction can generate 3D shapes faster, but the selection and processing of images may be a problem [27]. Therefore, combining edge computing and deep learning to achieve real-time 3D reconstruction of a single image may be a solution. In addition to the direct use of voxel methods to generate 3D shapes, other studies have used different three-dimensional representations, such as point clouds [28–30], meshes [31–33], primitives [34, 35], and implicit surfaces [36, 37]. Most of these methods can reconstruct three-dimensional objects with high resolution and are not limited by memory requirements. However, most of these methods need to solve the inherent defects of the model, such as using the point cloud method to reconstruct the surface details of the object and solving the genus problem of the mesh method to reconstruct the object.

For the voxel-based 3D object reconstruction method, it is robust to input. This method has the ability to adapt to 3D CNN and generate arbitrary topological structures. However, this method requires a huge amount of memory and calculations, and these factors make the resolution of the generated 3D shape too low. Therefore, how to solve the drawbacks of voxel-based 3D reconstruction is a premise for this method to generate high-resolution 3D shapes. At present, there are several methods for generating high-resolution 3D objects using voxel-based methods. As mentioned above, one of the methods is to use the sparse three-dimensional representation of the octree to generate high-resolution 3D shapes. It is also a method to transfer high-resolution 3D shape reconstruction to 2D space for implementation. Specifically, the method first uses the traditional 2D encoder-3D decoder architecture to generate a 3D object with low resolution from the input image. Then, superresolution reconstruction is performed on the 2D depth images of the low-resolution 3D object. Finally, the generated superresolution depth images are used for the reconstruction of a single high-resolution 3D object. In order to avoid directly manipulating voxels in a three-dimensional space, Richter and Roth [38] first predicted 6 depth maps of a 3D shape. They are then fused into a single reconstructed 3D shape. Smith et al. [39] also adopted a similar idea in the proposed MVD. They first used an encoder-decoder network to reconstruct the low-resolution 3D volume of a single image. Then, six orthographic depth maps of the low-resolution 3D object are obtained for superresolution reconstruction. Finally, the generated superresolution images are used to carve the upsampled low-resolution 3D shape to generate a high-resolution 3D object. This method can quickly accomplish high-resolution 3D object reconstruction of a single image.

However, the MVD method uses a traditional encoder-decoder network to generate low-resolution 3D shapes. This method has limited ability to extract image features in the 2D encoding stage, and the decoding speed in the 3D decoding stage is slow. In addition, the residual blocks (RB) used by MVD in depth image superresolution reconstruction do not fully utilize the features of different layers. This paper studies and improves these aspects to enhance the overall 3D reconstruction capabilities of the model. First, we improve the 2D encoder in the low-resolution 3D generation network into a 2D encoder with multifeature fusion to enhance the image feature extraction capability of the model. Then, we extend 2D ESPCN [40] to 3D ESPCN in the decoder stage to increase the speed of the decoder to generate 3D shapes. Second, this paper first introduces a single residual dense network (SRDN) on the basis of the residual network and dense network to improve. The residual network is then improved in a densely connected manner to maximize the reuse of features. Then, we obtain a multiresidual dense network (MRDN) to enhance the depth map superresolution network, which makes the network structure deeper and maximizes the information transfer between different convolutional layers. The experimental results show that the improved multiview decomposition (IMVD) structure performs better. First, the decoder using 3D ESPCN can increase the decoding speed of the model without degrading the performance of the model. Second, when the number of MRDB network layers is doubled compared to the number of RB network layers, the total model parameters and size are reduced by approximately 25%, respectively. Then, when the reconstructed object is in a relatively thin part, the reconstruction results of the MVD method are often broken. But our IMVD method can avoid this situation to some extent. In addition, the network that combines MFF and MRDB can capture more local features. The following sections are organized as follows. In Related Work, the current work related to this research is introduced. In Method, the improved MRDB and the low-resolution 3D object reconstruction network are introduced, respectively. In Experiment, the experiment is introduced, which includes the establishment of the dataset, the details of the training, and the relevant experimental results of each improvement component. In Conclusion, this paper is summarized.

The main contributions of this paper are summarized as follows:

(i) We propose an image encoder with multifeature fusion, which extracts the feature information of each layer to enhance the representation of the local details of the 3D shape. Compared with the traditional image encoder, the encoder with MFF is relatively more advantageous in capturing the detailed parts of 3D objects

(ii) We propose a 3D ESPCN operation to improve the traditional 3D decoder based on voxel representation, which reduces the time for the model to generate 3D shapes. Using 3D ESPCN can generate 3D shapes in lower resolution 3D volume spaces than traditional 3D decoders in the last step of the 3D decoding stage. This reduces the time required for the model to generate 3D shapes

(iii) We propose a multiresidual dense network to make full use of the features extracted from the residual network and the dense network. We connect the residual network in a dense manner and send the extracted features into the densely connected network. Model expression ability is improved by maximizing the reuse of features of each layer

2. Related Work

The goal of our work is to enhance its ability to generate high-resolution 3D objects from a single RGB image by improving the original MVD network. Wu et al. [18] earlier proposed the use of neural networks to recover the 3D shape of objects from 2.5D depth maps. Girdhar et al. [19] proposed a TL-embedding network. The network can complete the reconstruction from the RGB image to the 3D shape after training. These studies all apply a traditional encoder-decoder architecture, which uses progressive 2D convolution and 3D deconvolution for processing. Smith et al. [39] also used a similar structure to generate 3D shapes from 2D images. As we all know, in 2D image processing, the network layer that is too deep will cause the problem of gradient dispersion. When a network that is too deep can converge, its accuracy will also degrade. However, the deeper the network has also been proven to improve its performance. Therefore, it is an instinctive idea to introduce residual learning in the 3D reconstruction of a single image. Inspired by the residual network [41], Choy et al. [20] introduced a residual structure to design a deeper 3D object generation network. Their experimental results show that the network has a lower loss value in the training stage and can generate better 3D shapes than traditional 3D object generation networks. Similarly, Wu et al. [42] applied a similar residual structure in the 2D encoder. In addition, Soltani et al. [43] merged the residual block into the network to improve the performance of the model.

In the image superresolution, Dong et al. [44] first used convolutional neural networks to achieve superresolution reconstruction of low-resolution images. The input of this method is a high-resolution image after upsampling the low-resolution image. This superresolution method is complicated in operation and has a large amount of calculation. Subsequently, Shi et al. [40] proposed ESPCN. Different from upsampling input images to target resolution images for processing, they first use neural networks to extract features from low-resolution images. Then, the extracted features are recalculated using ESPCN operations to obtain high-resolution images. Since the feature extraction stage is performed on a lower resolution space, this method reduces the computational complexity of the entire superresolution process. Inspired by this, we first use a traditional 3D deconvolution operation to generate multiple low-resolution 3D volumes from the feature vector. Then, we expand ESPCN from 2D space to 3D space to generate a higher resolution 3D volume from these 3D volumes.

Recently, different network structures have appeared in image classification, such as the residual network (ResNet) [41] and the densely connected network (DenseNet) [45]. The purpose of introducing a residual network or densely connected network is to solve the problem of model degradation caused by designing a deeper network structure, and the deeper the network can extract more features to enhance the expression ability of the model. To reuse the feature information between more layers, a densely connected network is designed to solve the problem of gradient disappearance. Besides, the network structure designed in this way has a smaller model and requires less computation. Based on the above research, after analyzing the advantages and disadvantages of the residual block and the dense block, the Dual Path Network (DPN) [46] combines both to reduce the model parameters and to improve the training speed. Finally, better results were obtained in image classification, object detection, and semantic segmentation experiments. The relevant experimental results show that different structures have different benefits to the performance, parameter size, and computational complexity of the model.

Later on, various extended feature extraction structures were gradually introduced in the experiment of image superresolution reconstruction [47], such as the deep residual recurrent network (DRRN) [48] and the residual block [49]. In the superresolution experiment of 2D images, a multilayer feature concatenation method is often introduced to obtain more image feature information. Zhang et al. [50] proposed a residual dense network (RDN) after studying the residual block and the dense block. The output of each residual dense block (RDB) is processed through local feature fusion and global feature fusion. They further explore how to make full use of the features of different convolutional layers through this multifusion method. Wang et al. [51] introduced the residual-in-residual dense block (RRDB) to connect different network layers to make the model achieve better performance. Inspired by these studies, we study a multiresidual dense block to make full use of the features of each convolutional layer.

3. Method

In this section, we introduce an improved multiview decomposition (IMVD) network, as shown in Figure 1. The goal of this paper is to improve the MVD network to enhance the expression ability of the model and raise the quality of 3D object reconstruction. In the following content, we first describe the improved multiresidual dense block (MRDB) network. Second, a 2D encoder with multilayer feature fusion is described. Finally, we briefly introduce the 3D subpixel convolutional layer (3D SPCL) in 3D ESPCN.

3.1. Multiresidual Dense Network. The depth map superresolution network of MVD is based on the residual block in the

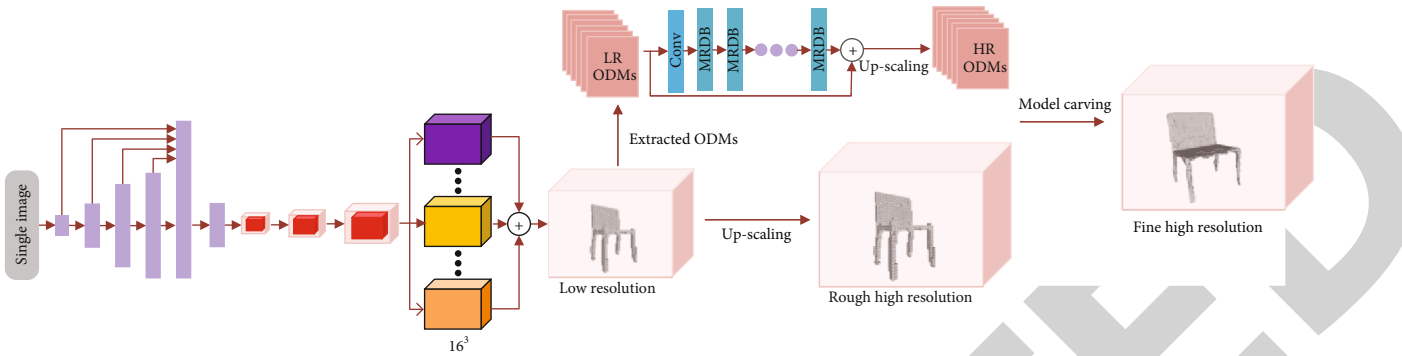


Figure 1: Improved single-RGB image high-resolution 3D reconstruction network structure. We apply the basic architecture of MVD [39]. The improved multiresidual dense block superresolution network processes six axis-aligned orthographic depth maps (ODMs). High-resolution depth maps and silhouette maps are estimated from the low-resolution ODMs, respectively. The improved multifeature fusion 2D encoder encodes the input RGB image into a 1024-dimensional latent vector and decodes it through 3D ESPCN.

generator of SRGAN [49]. Our improved superresolution network is based on a combination of the residual network and dense network. This improvement is to increase the connections between the convolutional layers to obtain more feature information and to design deeper and more complex structures.

Recent experiments have shown that connecting more layers in a network structure can further improve the performance of the model. Similarly, the use of denser connections in 2D images has also proved to enhance the performance of the model. Chen et al. [46] demonstrated that a single residual network has less redundancy in reusing features, and this shared information strategy makes it difficult to learn new features. However, a single densely connected network will lead to high redundancy while learning multiple new features. Finally, they designed a DPN with the advantages of the residual network and the densely connected network. In addition, Zhang et al. [50] also explored the combination of the residual network and the dense network. Their experimental results showed that the combination of both is beneficial. Similarly, we also take both into consideration. First, we introduce a single residual dense block (SRDB) [50]. Then, we improve on the basis of a single residual dense block and design a new multiresidual dense block (MRDB) by connecting the residual learning in a dense manner, as shown in Figure 2.

The MVD basic architecture uses sixteen residual blocks as shown in Figure 2(a). We maintain the basic architecture of MVD. We apply L multiresidual dense blocks as shown in Figure 2(c). The basic structure of the multiresidual dense network is shown in Figure 1. First, we consider a single image x_0 as the input of the superresolution network. Each layer of the network input consists of one or more components: batch normalization (BN) and convolution (Conv), and we represent these nonlinear transformations as $H_l(\cdot)$, where l indexes the layer. Then, $H_l(\cdot)$ in Figure 2 is in the form of Conv-BN-Conv-BN. Then, T denotes a transition layer consisting of a 1×1 convolution layer and batch normalization.

3.1.1. ResNet. Compared with the traditional CNN, inserting shortcut connections between different convolutional layers can convert it into a residual network, as shown in Figure 2(a). When the input and output dimensions of different convolutional layers are the same, the identity shortcut connection can be used to directly add its output to the output of the subsequent layer. When using the identity shortcut connection method, this connection method neither adds new parameters nor increases the computational complexity. For the residual network of Figure 2(a), the output x_{l-1} from the $(l-1)$th layer bypasses the nonlinear transformations with an identity function, and the results are added as the lth layer input. The residual network can be expressed as follows:

$$x_l = H_l(x_{l-1}) + x_{l-1}. \tag{1}$$

3.1.2. Single Residual Dense Network (SRDN). ResNet uses shortcut connections to solve the problem of model degradation to a certain extent. However, the connection between different layers of ResNet is a sparse connection. In order to make full use of the features of different layers, DenseNet uses the output of each layer as the input of each subsequent layer. This densely connected approach allows the model to achieve better performance than ResNet with fewer parameters and computational costs. In the single residual dense block of Figure 2(b), the input of the lth layer is derived from the output features of the previous 0th, 1th, $\cdots$, $(l-1)$th layers, x_0, $x_1, \cdots, x_{l-1}$:

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]), \tag{2}$$

where $[x_0, x_1, \cdots, x_{l-1}]$ represents the concatenation operation. Equation (2) is also known as densely connected network output. Finally, a SRDB result consists of the input x_0 summed with the T output by a shortcut connection. We call this network SRDN, and its output can be expressed as

$$x_{\text{SRDB}} = T(x_l) + x_0. \tag{3}$$

3.1.3. Multiresidual Dense Network (MRDN). In each SRDB, DenseNet is applied to extract the features of different layers for fusion, and single residual learning is introduced to improve the information flow. It should be noted that residual learning in SRDB is not closely combined with DenseNet. In order to further improve the information flow, we fuse the residual learning of different layers with DenseNet. Now we

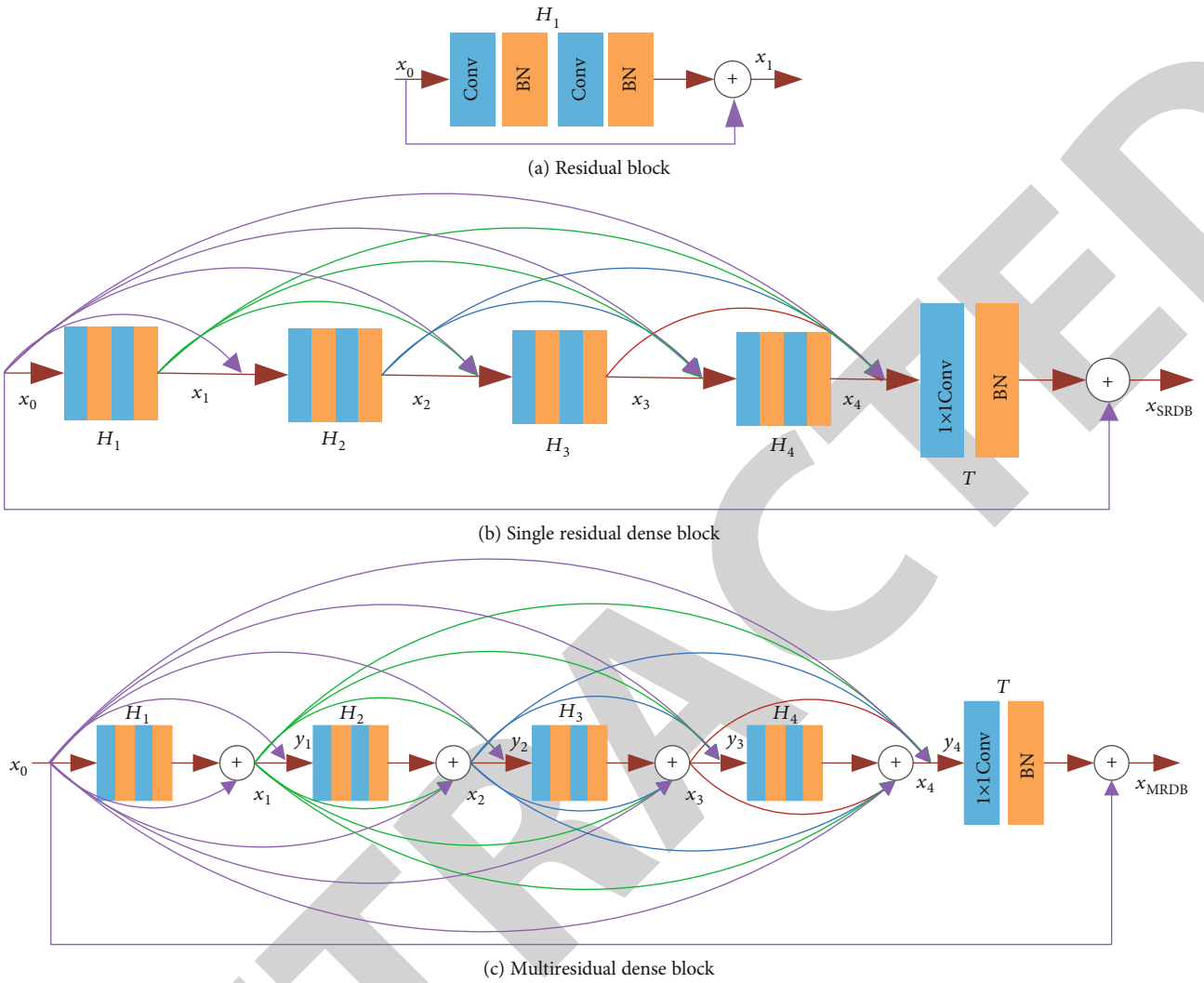


Figure 2: (a) Residual block in MVD [39]. (b) Single residual dense block in RDN [50]. (c) Our multiresidual dense block.

consider the multiresidual dense block of Figure 2(c). First, we denote the x_0 and y_0 as the residual input and dense input of a single MRDB, and $x_0 = y_0$. For x_1, it can be expressed as

$$x_1 = H_1(y_0) + x_0. \tag{4}$$

Then, y_1 is expressed as the fusion of residual output x_1 and x_0:

$$y_1 = [x_0, x_1]. \tag{5}$$

Combining Equations (4) and (5), it can be seen that the input of DenseNet in MRDB includes the output of RenseNet.

Further, we denote that x_l and y_l are the output of the residual network and the densely connected network in the lth layer, respectively. The lth layer accepts all of the preceding input feature maps $x_0, x_1, \cdots, x_{l-1}$ and the y_{l-1} of the $(l-1)$th layer as the residual output x_l:

$$x_l = H_l([y_{l-1}]) + \left(\sum_{t=0}^{l-1} x_t\right). \tag{6}$$

Similarly, we can get the output y_l of the lth layer:

$$y_l = [x_0, x_1, \cdots, x_l]. \tag{7}$$

Thus, transform Equation (7) into Equation (6), and Equation (6) can be further written as

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]) + \left(\sum_{t=0}^{l-1} x_t\right). \tag{8}$$

Comparing Equation (8) with Equation (2), the first term on the right side of Equation (8) is formally equal to Equation (2). However, $x_1, \cdots, x_{l-1}$ in Equation (8) is essentially the residual input of Equation (1). In addition, Equation (8) adds a summation operation for all feature maps $x_0, x_1, \cdots, x_{l-1}$ of the preceding lth layer. From the above analysis, Equation (8) combines the features of the residual network and the dense network and expands them.

Finally, the output of a single MRDB can be expressed as

$$x_{MRDB} = T(y_l) + x_0. \tag{9}$$

We assume that the growth rate of the model is G [45]. Each $H_l(\cdot)$ produces G feature maps, and the result is $G_0 + G \times (l-1)$, where G_0 is the number of feature map channels of the input layer.

3.1.4. Implementation Details. We use the structure shown in Figures 2(b) and 2(c) in single residual dense networks and multiresidual dense networks, respectively. In the experiment, the kernel filter stride length of all convolutional layers is 1. The kernel depth G is 128 and 64, respectively. Since the multiresidual dense network has deeper and denser connections, it will inevitably lead to an increase in the parameters of the model. Performing 1×1 convolution after feature input is a common means of reducing model parameters [45, 50]. Our $H_l(\cdot)$ form is Conv(1×1)-BN-Conv(3×3)-BN. In addition, the final concatenation operation of each multiresidual dense block produces a large number of feature maps. We use 1×1 convolution to reduce its number and follow a batch normalization operation to feed the next multiresidual dense block. We let the number of single residual dense blocks and multiresidual dense blocks be L, which is set to 8 or 4 in the experiment.

3.2. Low-Resolution Network. The bottom of Figure 1 shows the overall low-resolution 3D reconstruction network. First, a 2D encoder with multifeature fusion is used to encode the input image into a fixed-length hidden layer vector. Then, traditional 3D deconvolution and 3D ESPCN are used to decode the latent vector to generate a low-resolution 3D volume. In the next part, we will introduce the 2D encoder with multifeature fusion and 3D ESPCN, respectively.

3.2.1. 2D Encoder with Multifeature Fusion. For coarse-to-fine 3D object reconstruction methods, high-quality low-resolution 3D object reconstruction is a basis for its higher resolution 3D reconstruction. In order to further improve the feature extraction capability of the 2D encoder to enhance the 3D reconstruction performance of the model, we use different layers of feature maps for fusion. An improved network comparison is shown in Figure 3.

Both encoder networks consist of a standard convolutional layer, a batch normalization layer, and a leaky rectified linear unit (LReLU). The encoder encodes the input data into a low-dimensional hidden vector, and the decoder decodes the compressed vector to reconstruct a 3D object. The advantage of this approach is that it can compress the input high-dimensional data into a low-dimensional representation and then reconstruct its 3D object through the representation.

By observing the traditional encoder of Figure 3(a), we find that the encoder of this mode has less utilization of features. In the image superresolution experiment of RDN [50], the global feature fusion (GFF) method proved to be able to improve the performance of the model. This is a method of extracting the output of all residual dense blocks in the network for fusion. Inspired by this, we extract the output from each nonlinear transformation $H_l(\cdot)$ in the encoder to fuse, as shown in Figure 3(b). To match the number of $H_l(\cdot)$ output feature map channels of different lth layers, we use a 1×1 convolution. The definition of $H_l(\cdot)$ is consistent with Section 3.1. Since the number of convolution channels after feature fusion is too large, their direct compression to a 1024-dimensional feature vector will result in huge model parameters. Therefore, we use a 1×1 convolution to reduce the dimensions of the fused features. The multifeature fusion encoder output is expressed as

$$x_{\mathrm{MFF}} = T([x_1, x_2, \cdots, x_l]). \tag{10}$$

Finally, the output of the encoder is compressed to a 1024-dimensional feature vector through a flat layer and a fully connected layer. We find that multilayer feature fusion can encourage models to learn new features.

3.2.2. 3D Subpixel Convolution Layer. In the image superresolution experiment, combining multiple low-resolution images (feature maps in low-resolution space) to generate a higher resolution image is a more efficient processing method [40]. Inspired by this, in the voxel-based 3D convolutional neural network, multiple low-resolution 3D shapes can be combined into a higher resolution 3D shape. This operation can be named 3D SPCL, as shown in Figure 4.

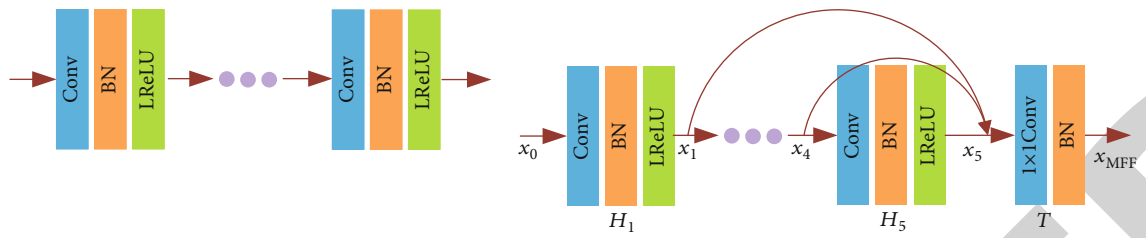
Generally, the size of a single low-resolution 3D volume and a single high-resolution 3D volume can be expressed as $H \times W \times D$ and $nh \times nW \times nD$, respectively. We will refer to n as the upscaling ratio. First, a traditional voxel-based decoder is used to generate n^3 low-resolution 3D shapes from the latent space, the size of which is $H \times W \times D \cdot n^3$. Then, 3D SPCL is used to rearrange the generated n^3 low-resolution 3D shapes into one high-resolution 3D shape. 3D SPCL is a periodic operation that rearranges the elements of the $H \times W \times D \times n^3$ tensor to a tensor of shape $nH \times W \times D \cdot n^2$. Then, the W channel and the D channel are arranged in sequence. Finally, a tensor of shape $nH \times nW \times nD$ is the output. The entire 3D SPCL does not involve convolution operations. Compared with the traditional 3D decoding method based on voxels, this method reduces the 3D deconvolution operation at higher resolution. Therefore, using 3D SPCL when generating 3D shapes can make the model have a faster decoding speed.

4. Experiment

In this part, we show the experimental results of the improved multiview decomposition (IMVD) network for 3D object superresolution and 3D object reconstruction of a single RGB image. In addition, we analyze the importance of each component in the network. The qualitative and quantitative results show that the proposed method can improve the expression ability of the model.

4.1. Dataset and Metric

4.1.1. 3D Object Superresolution Dataset. The 3D object superresolution dataset consists of a $32 \times 32 \times 32$ low-resolution voxel model and a corresponding $256 \times 256 \times 256$ high-resolution voxel model. Following the MVD approach, we also use the ShapeNetCore [52] dataset to transform CAD models into 3D shapes represented by voxels. Two classes are selected from the ShapeNetCore dataset: chair and plane. Their



(a) Traditional encoder network (b) Encoder network with multifeature fusion

Figure 3: Improved network structure for comparison. (a) Traditional encoder network in MVD [39]. (b) Our encoder network.

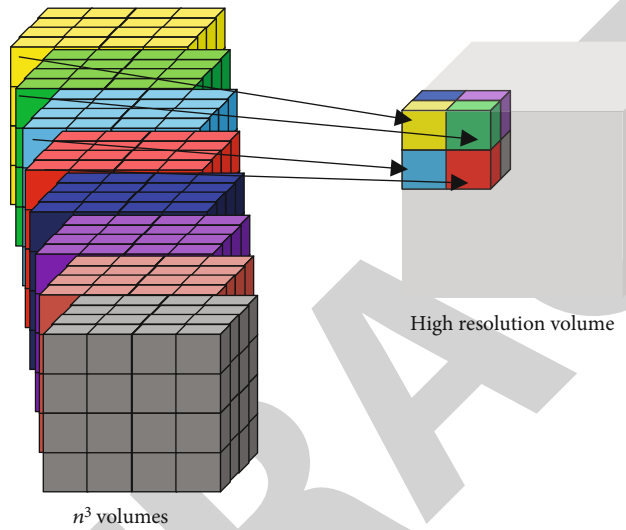


Figure 4: The 3D SPCL operation rearranges multiple low-resolution 3D volumes into a high-resolution 3D volume.

numbers are approximately 7000 and 4000, respectively. We preprocess the 3D object superresolution dataset and extract 6 orthographic depth maps (ODMs) for each object in the dataset corresponding to low resolution and high resolution. The final dataset is divided into a training set, a validation set, and a test set. We used 70% of the dataset as the training set, 10% as the validation set, and 20% as the test set. The dataset we created is named 3D superresolution dataset ($Data_{SR}$).

4.1.2. Low-Resolution 3D Reconstruction Dataset. The 3D object reconstruction experimental dataset of a single RGB image is based on $Data_{SR}$. Similarly, we refer to the relevant dataset production methods in MVD. Based on the completed $Data_{SR}$, we render each CAD model as a 128×128 RGB image to obtain a random viewpoint and possible azimuthal rotation of the object between $(-20°, 30°)$. Similarly, the completed dataset is divided into a training set, a validation set, and a test set according to the 3D superresolution experimental dataset, with a ratio of 70 : 10 : 20, respectively. Finally, the dataset we follow is named $Data_{HSP}$.

4.1.3. Evaluation Metric. In all 3D reconstruction experiments, the evaluation metric uses the intersection over union (IoU). Applying IoU to evaluate the corresponding model on the $Data_{SR}$ and $Data_{HSP}$ enables quantitative analysis of model performance.

4.2. Training Details. We train the entire model in two stages. The 3D superresolution model and the low-resolution 3D reconstruction model are separately trained. Finally, the two training models of the two stages are combined to form the final high-resolution 3D object reconstruction model of a single RGB image, which is the improved multiview decomposition (IMVD) network.

In the 3D object superresolution experiment, the silhouette estimation network and the depth estimation network are, respectively, trained. Following the MVD, the 3D object superresolution experiment was reconstructed from $32 \times 32 \times 32$ resolution to $256 \times 256 \times 256$ resolution. The dataset used for model training comes from the 3D superresolution dataset described in Section 4.1. During the training process, both use the Adam [53] default parameter training, the learning rate is 10^{-4}, the training minimum batch size is 32, the training epoch is 300, and the error function uses the mean square error (MSE) loss function. The training set is used for network training, and the validation set is used to evaluate model performance at the end of each epoch. The current model is retained only if the IoU score of the reconstruction

result evaluation is greater than the largest IoU score of the previous reconstruction result.

In a low-resolution 3D object reconstruction experiment, the encoder with multifeature fusion and the 3D ESPCN decoder are trained. Using the Adam optimizer, the learning rate is 10^{-3}, the training minimum batch is 128, the training epoch is 300, and the mean square error term is used as the loss function. The update of the model is the same as the operation in the 3D object superresolution experiment.

After the silhouette estimation network, the depth estimation network, and the low-resolution 3D object reconstruction network have all been trained, the 3D model carving combines three networks to accomplish the high-resolution reconstruction. For model carving, it includes silhouette carving and depth map carving. Firstly, the rough 3D shape after upsampling is carved using estimated silhouette maps to ensure the correctness of its structure. Then, the estimated depth maps will be used for detail carving. The voxels that have not reached the corresponding depth in the 3D shape after silhouette carving will be deleted. We implemented the model with the TensorFlow Architecture and trained on a single NVIDIA GTX 1080 GPU.

4.3. 3D Object Superresolution Experiment

4.3.1. Model Parameters, Size, and IoU Comparison. Table 1 shows the experimental comparison of SRDN and MRDN on the Data_{SR} chair for different block numbers L (8 or 4) and different size feature maps G (128 or 64). The number in italic in Table 1 indicates the highest IoU score for the corresponding category of 3D reconstruction. We use SRDN and MRDN to improve MVD in superresolution experiments of the chair and can achieve higher IoU scores than MVD. We roughly calculate the number of MVD superresolution network layers with 16 residual blocks as shown in Figure 2(a), and the total number of layers is 32. Similarly, the number of IMVD network layers improved by MRDB is 72.

As can be seen from Table 1, when the number of network layers is increased by about 1 time, the MRDN model parameters are reduced by about 25%. At the expense of the IoU reconstruction score, the model parameters are reduced by 81% when the feature map G is reduced by half. We observe that in the MRDB experiment, keeping the feature map G constant and reducing L by half make the model IoU fall. This suggests that designing deeper networks can enhance the expressive ability of the model. In Table 1, MRDN-4 ($G = 128$) and MRDN-8 ($G = 64$) are scaled-down on L and G, respectively. Although the IoU scores are almost the same, the latter model parameters are reduced by approximately 56%. In addition, the MRDN model parameters can be reduced by 45% when SRDN and MRDN are close to the obtained IoU score.

4.3.2. Qualitative Results. We show qualitative results in Figure 5. We rendered from 32^3 resolution to 256^3 on the test set. The low-resolution 3D shapes of real chairs and planes are used as input for this experiment (line 1 of Figure 5). The output results of MVD [39] are shown in line 2 of Figure 5. The IMVD results are shown in line 3 of Figure 5.

Table 1: Comparison of parameters and IoU (%) on the Data_{SR} chair model. "-" means that the model is out of our running memory without IoU results. "*" indicates the result of our implementation.

Method	Parameters	Size	IoU
RB [39]	5.28M	21.1M	68.4*
MRDN-4 ($G = 128$)	2.25M	9.0M	69.3
SRDN-8 ($G = 64$)	1.83M	7.3M	69.1
MRDN-8 ($G = 64$)	1.00M	4.0M	69.2
SRDN-8 ($G = 128$)	7.27M	-	-
MRDN-8 ($G = 128$)	3.97M	15.9M	69.8

As can be seen from the comparison of Figure 5, the MVD method tends to break in a thin object portion. However, our IMVD results are more complete in this situation. The experimental results show that extracting more feature information through the multiresidual dense network is beneficial to enhance the expressive ability of the model.

4.3.3. Quantitative Results. We trained each class in Data_{HSP} separately in a 3D object superresolution experiment. The results are compared with various methods employed in MVD and presented in Table 2. The benchmark method directly increases the resolution of the 3D volume from 32^3 to 256^3 through the nearest neighbor upsampling. The MVD method combines depth estimation and silhouette estimation. It can be seen from Table 2 that our method performs better than the MVD method in the experiment. We all achieved higher scores in different categories.

4.4. Single-Image 3D Reconstruction Experiment

4.5. Model Parameters and Iteration Time. We show the parameter sizes and required iteration time of different low-resolution 3D reconstruction models, as shown in Table 3. It can be seen from Table 3 that IMVD has increased in the number of parameters and decreased in iteration time. Generally, 3D reconstruction experiments of a single image often use 13 categories in the ShapeNetCore dataset. The total number of models in 13 categories is approximately 39,832. According to the method of generating the dataset in this article, the number of models in the training set of each category is approximately 2,144. According to the iteration time in Table 3 and the training method in this paper, the training time of IMVD in 13 categories will be reduced by approximately 4 hours compared with MVD. For higher resolution 3D reconstruction experiments, this method has more advantages in training time.

4.5.1. Convergence Curve Analysis. In Figure 6, we show the convergence curve on the validation set. In Figures 6(a) and 6(b), the red curves represent the convergence of the MVD method on the chair and aircraft validation set, respectively. Similarly, the green curve corresponds to our IMVD method. We train the model to use the same parameters, just changing the structure of the model. The training epoch was 300, and the reconstructed IoU score was evaluated on the validation set at the end of each epoch. The original MVD oscillated

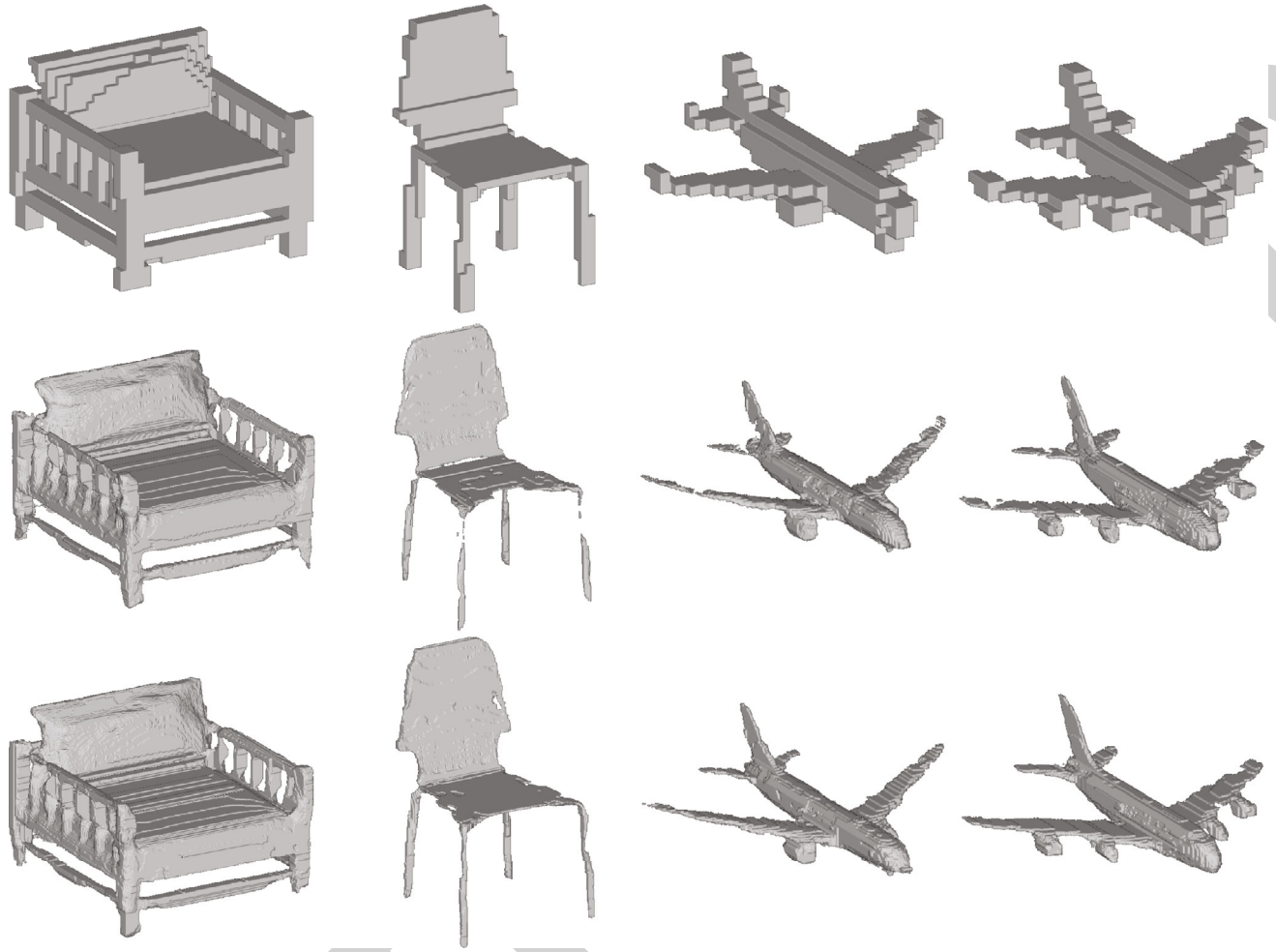


Figure 5: 3D object superresolution results on $Data_{SR}$.

Table 2: 3D object superresolution reconstruction IoU score at 256^3.

Class	Benchmark [39]	Depth [39]	Silhouette [39]	MVD [39]	IMVD (ours)
Chair	54.9	58.5	67.3	68.5	69.8
Plane	39.9	50.5	70.2	71.1	72.9

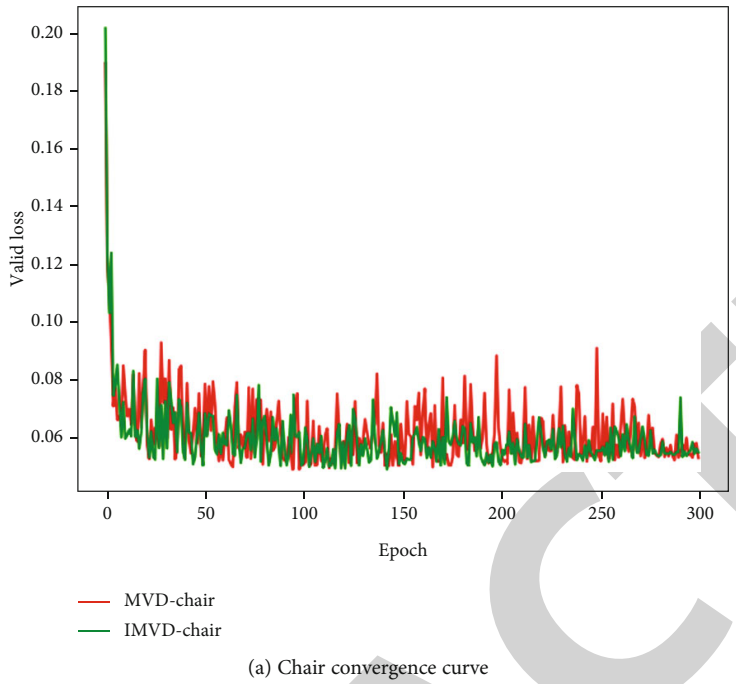
Table 3: Model parameters and iteration time at 32^3 resolution. The batch size is 2.

Method	Parameters (M)	Iteration time (ms)
MVD [39]	27.02	50.8
MVD+MFF	27.15	49.9
MVD+3D ESPCN	27.01	47.7
IMVD	27.14	47.1

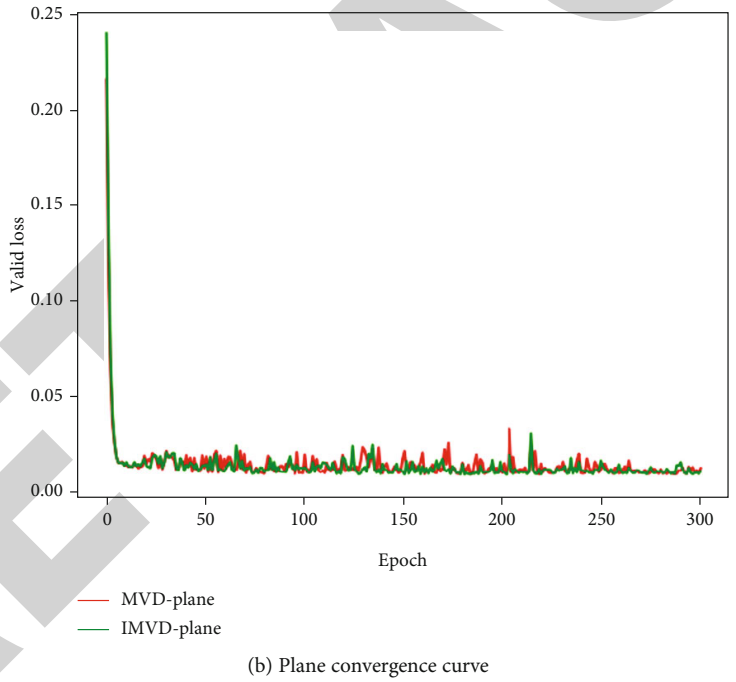
over the entire training cycle of the training chair. Our IMVD uses a multifeature fusion approach to reduce the degree of model oscillation, which helps to improve the model expression ability. In Figure 6(b), the model of the aircraft itself has no complicated and thin parts like a chair. Therefore, it seems that there is not much difference between the improved convergence curves of the IMVD network and the original MVD network on the validation set. In summary, we can see from the comparative analysis in Figure 6 that the improved network can improve the stability of model training.

4.5.2. Quantitative Results. We show quantitative results in Table 4. We compared several methods, HSP [22], AE [39], and MVD [39], which all use $Data_{HSP}$ to reconstruct 3D objects from a single RGB image at 256^3 resolution. As can be seen from Table 4, the proposed IMVD method can achieve a higher IoU score on a single-image reconstruction 256^3 resolution 3D object.

4.6. Ablation Studies. Table 5 quantitatively demonstrates the effects of MFF, 3D ESPCN, and MRDB. The IoU scores of the reconstruction results are in the second column, and the third column corresponds to the plane and the chair, respectively. The last column represents the average IoU score for the plane and chair reconstruction results. The first column in Table 5 represents the combination of the different components we proposed. Among them, the benchmark is the method of MVD. We add MFF and MRDB (from line 3 to line 4 of Table 5) to the benchmark method. Since the



(a) Chair convergence curve



(b) Plane convergence curve

Figure 6: Convergence curve analysis on the validation set. The curve represents the evaluation of the loss value over 300 epochs of the corresponding validation set on the Data_{HSP}.

Table 4: Single-image reconstruction IoU score at 256^3 resolution.

Class	AE [39]	HSP [22]	MVD [39]	IMVD (ours)
Chair	36.4	37.8	40.1	41.9
Plane	28.6	56.1	56.4	58.8

Table 5: The IoU score evaluates the contribution of each component.

Component	Chair	Plane	Average
Benchmark [39]	40.1	56.4	48.25
3D ESPCN	40.2	56.4	48.30
MFF	41.2	57.9	49.55
MRDB	41.3	57.0	49.15
MFF+MRDB	41.9	58.6	50.25

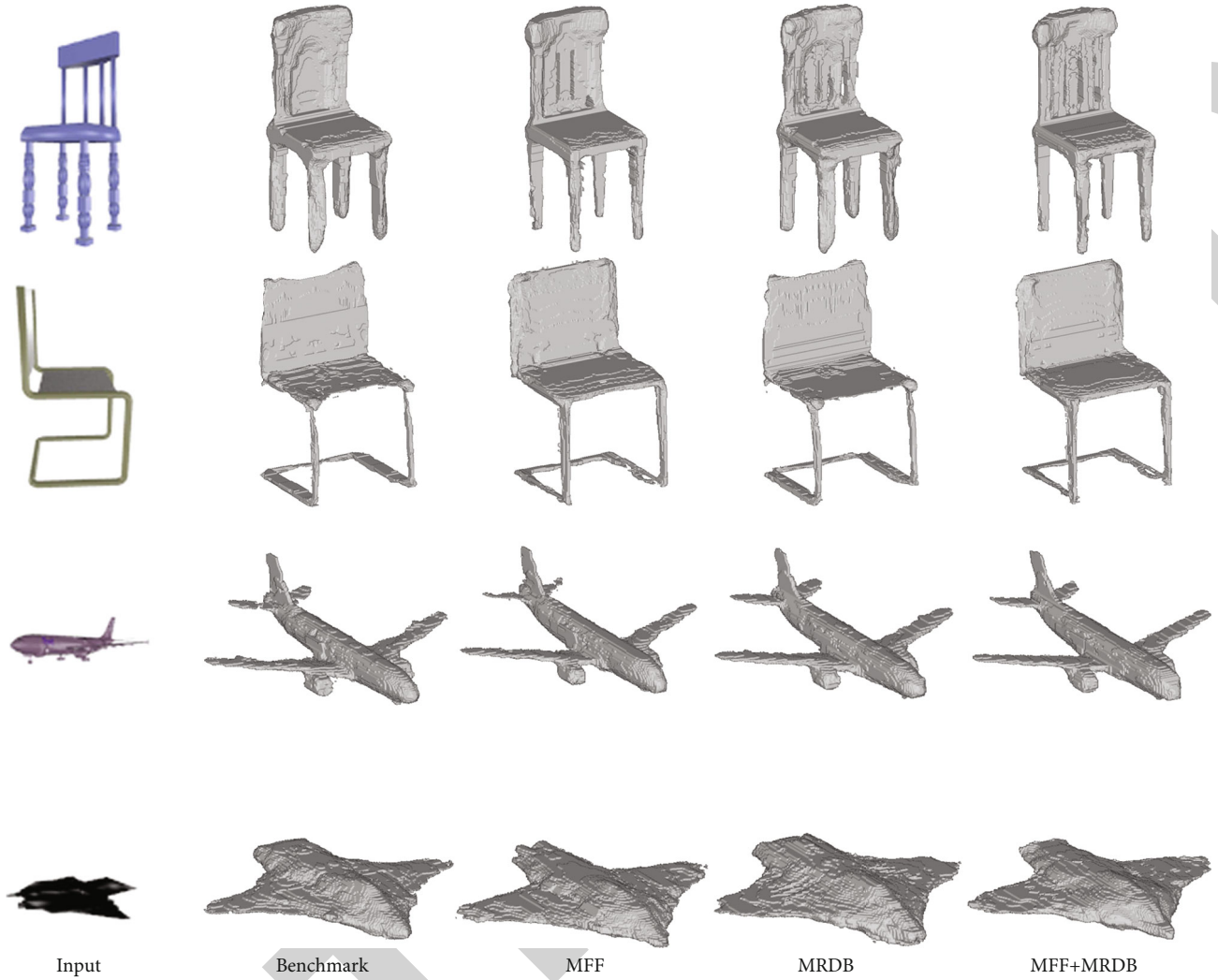


Figure 7: Qualitative results of the ablation study. This figure reflects the contribution of each improved component to the model.

addition of 3D ESPCN basically did not improve the performance of the model, it can be seen that adding another component can improve the performance of the model. We add modules for the combination of MFF and MRDB on the benchmark (in the last row of Table 5). After adding two components, the performance of the model has been further improved.

Figure 7 qualitatively shows the contribution of MFF and MRDB in the model. The first column of Figure 7 represents the input RGB image. The second column is a method of MVD, and the reconstruction result is broken at the edge portion (columns 3 to 5 of Figure 7). However, partial fractures have been improved after the addition of MFF or MRDB. In addition, it can be seen in the reconstruction of the first row of the chair in Figure 7 that the input RGB image of the chair back is a series of unconnected pillars. However, the 3D reconstruction result of MVD does not reflect this feature. After adding MFF or MRDB alone, the reconstruction results show this part of the details. This detail can be further enhanced after combining MFF and MRDB. It can be seen from the comparison of the third column to the fifth column of Figure 7 that the final reconstruction result of IMVD is mainly refined based on MFF. This also reflects the impact of the resolution of low-resolution 3D object reconstruction on high-resolution 3D object representation. At present, the rendering of CAD models in the dataset is performed in random colors, and the background of all rendered images is clean. In the future, images with textures and backgrounds can be used for rendering to enrich the dataset, which will make the model more robust to 3D object reconstruction from 2D images in real scenes. In addition, there are other methods, such as exploring new algorithms to extract more effective image features, using different training architectures, and supervising methods to optimize [54].

5. Conclusion

We improve the depth map superresolution network and low-resolution 3D reconstruction network of the single image in MVD, respectively. The improved model shows better performance compared with MVD in the corresponding experiment. We propose an architecture that includes multiple MRDB blocks, which can make the network structure design deeper and make full use of the multilayer

structure information to enhance the model expression ability. Even though the network design is deeper, the model parameters are even smaller. In addition, we use multifeature fusion and 3D ESPCN to improve the 2D encoder and 3D decoder, respectively. Both of these can reduce the training time of the model. At present, there are few studies on 3D reconstruction technology and edge computing based on deep learning, but their combination has broad application prospects. In intelligent manufacturing, edge computing is conducive to extend various computing resources to the edge of the Internet of Things and realizes manufacturing and production [55]. However, the problem of 3D data heterogeneity between different devices may need to be resolved. The use of 3D reconstruction methods based on deep learning may be one of the means to solve this problem in the future.

Data Availability

The 3D model dataset used to support the findings of this study can be downloaded from the public website: https://www.shapenet.org/.

Conflicts of Interest

No potential conflict of interest was reported by the authors.

Authors' Contributions

Jiansheng Peng, Kui Fu, and Qingjin Wei contributed equally to this work.

Acknowledgments

The authors are highly thankful to the National Natural Science Foundation of China (NO. 62063006), to the Development Research Center of Guangxi Relatively Sparse-populated Minorities (ID: GXRKJSZ201901), and to the Natural Science Foundation of Guangxi Province (NO. 2018GXNSFAA281164). This research was financially supported by the project of outstanding thousand young teachers' training in higher education institutions of Guangxi, Guangxi Colleges and Universities Key Laboratory Breeding Base of System Control and Information Processing.

References

[1] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, Las Vegas, NY, USA, 2016.

[2] K. Haming and G. Peters, "The structure-from-motion reconstruction pipeline-a survey with focus on short image sequences," *Kybernetika*, vol. 46, no. 5, pp. 926–937, 2010.

[3] C. Cadena, L. Carlone, H. Carrillo et al., "Past, present, and future of simultaneous localization and mapping: toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[4] L. Galteri, C. Ferrari, G. Lisanti, S. Berretti, and A. Del Bimbo, "Deep 3D morphable model refinement via progressive growing of conditional generative adversarial networks," *Computer Vision and Image Understanding*, vol. 185, pp. 31–42, 2019.

[5] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Category-specific object reconstruction from a single image," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1966–1974, Boston, MA, USA, 2015.

[6] Z. Yao, D. He, Y. Chen et al., "Inspection of exterior substance on high-speed train bottom based on improved deep learning method," *Measurement*, vol. 163, article 108013, 2020.

[7] 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*, vol. 32, no. 9, pp. 4387–4415, 2020.

[8] S. Wan, X. Xu, T. Wang, and Z. Gu, "An intelligent video analysis method for abnormal event detection in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2020.

[9] S. Wan, Y. Xia, L. Qi, Y.-H. Yang, and M. Atiquzzaman, "Automated colorization of a grayscale image with seed points propagation," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1756–1768, 2020.

[10] Y. Xi, Y. Zhang, S. Ding, and S. Wan, "Visual question answering model based on visual relationship detection," *Signal Processing: Image Communication*, vol. 80, article 115648, 2020.

[11] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, vol. 398, pp. 520–530, 2020.

[12] C. Zhang, X. Guo, X. Guo et al., "Machine learning model comparison for automatic segmentation of intracoronary optical coherence tomography and plaque cap thickness quantification," *Computer Modeling in Engineering & Sciences*, vol. 123, no. 2, pp. 631–646, 2020.

[13] S. Wan, Z. Gu, and Q. Ni, "Cognitive computing and wireless communications on the edge for healthcare service robots," *Computer Communications*, vol. 149, pp. 99–106, 2020.

[14] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Computer Networks*, vol. 168, article 107036, 2020.

[15] J. Chen, K. Li, Q. Deng, K. Li, and P. S. Yu, "Distributed deep learning model for intelligent video surveillance systems with edge computing," *IEEE Transactions on Industrial Informatics*, 2019.

[16] C. Liu, Y. Cao, Y. Luo et al., "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure," *IEEE Transactions on Services Computing*, vol. 11, pp. 249–261, 2018.

[17] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim, and C. S. Hong, "Deep learning based caching for self-driving cars in multi-access edge computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2020.

[18] Z. Wu, S. Song, A. Khosla et al., "3D ShapeNets: a deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, Boston, MA, USA, 2015.

[19] R. Girdhar, D. F. Fouhe, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Computer Vision – ECCV 2016. ECCV 2016*, pp. 484–499, Springer, 2016.

[20] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: a unified approach for single and multi-view 3D object

reconstruction," in *Computer Vision – ECCV 2016. ECCV 2016*, pp. 628–644, Springer, 2016.

[21] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: efficient convolutional architectures for high-resolution 3D outputs," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2088–2096, Venice, Italy, 2017.

[22] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *2017 International Conference on 3D Vision (3DV)*, pp. 76–84, Qingdao, China, 2017.

[23] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "OctNetFusion: learning depth fusion from data," in *2017 International Conference on 3D Vision (3DV)*, pp. 57–66, Qingdao, China, 2017.

[24] S. Yu, X. Chen, S. Wang, L. Pu, and D. Wu, "An edge computing-based photo crowdsourcing framework for real-time 3D reconstruction," *IEEE Transactions on Mobile Computing*, 2020.

[25] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 82–90, Barcelona, Spain, 2016.

[26] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[27] X. Xu, X. Liu, X. Yin, S. Wang, Q. Qi, and L. Qi, "Privacy-aware offloading for training tasks of generative adversarial network in edge computing," *Information Sciences*, vol. 532, pp. 1–15, 2020.

[28] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 605–613, Honolulu, HI, USA, 2017.

[29] K. L. Navaneet, P. Mandikal, M. Agarwal, and R. V. Babu, "CAPNet: continuous approximation projection for 3D point cloud reconstruction using 2D supervision," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8819–8826, Hilton Midtown, NY, USA, 2019.

[30] P. Mandikal and V. B. Radhakrishnan, "Dense 3D point cloud reconstruction using a deep pyramid network," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1052–1060, Waikoloa Village, HI, USA, 2019.

[31] H. Kato and T. Harada, "Learning view priors for single-view 3D reconstruction," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9778–9787, Long Beach, CA, USA, 2019.

[32] C. Wen, Y. Zhang, Z. Li, and Y. Fu, "Pixel2Mesh++: multi-view 3D mesh generation via deformation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1042–1051, Munich, Germany, 2019.

[33] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3D surface generation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 216–224, Salt Lake City, UT, USA, 2018.

[34] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem, "3D-PRNN: generating shape primitives with recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 900–909, Venice, Italy, 2017.

[35] P. S. Wang, Y. Liu, Y. X. Guo, C. Sun, and X. Tong, "O-CNN: octree-based convolutional neural networks for 3D shape analysis," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 72–81, 2016.

[36] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 165–174, Long Beach, CA, 2019.

[37] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann, "DISN: deep implicit surface network for high-quality single-view 3D reconstruction," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 495–502, Vancouver, Canada, 2019.

[38] S. R. Richter and S. Roth, "Matryoshka networks: predicting 3D geometry via nested shape layers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1936–1944, Salt Lake City, USA, 2018.

[39] E. Smith, S. Fujimoto, and D. Meger, "Multi-view silhouette and depth decomposition for high resolution 3D object representation," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 6479–6489, Montréal, Canada, 2018.

[40] W. Shi, J. Caballero, F. Huszar et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, Las Vegas, NY, USA, 2016.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

[42] J. Wu, Y. Wang, T. Xue, and X. Sun, "MarrNet: 3D shape reconstruction via 2.5D sketches," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 8–15, Long Beach, CA, USA, 2017.

[43] A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum, "Synthesizing 3D shapes via modeling multi-view depth maps and silhouettes with deep generative networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1511–1519, HI, USA, 2017.

[44] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of the European Conference on Computer Vision*, pp. 184–199, Cham, 2014.

[45] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, USA, 2017.

[46] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 4467–4475, Long Beach, CA, USA, 2017.

[47] K. Fu, J. Peng, H. Zhang, X. Wang, and F. Jiang, "Image super-resolution based on generative adversarial networks: a brief review," *CMC-Computers, Materials & Continua*, vol. 64, no. 3, pp. 1977–1997, 2020.

[48] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3147–3155, Honolulu, HI, USA, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8864533, 12 pages
https://doi.org/10.1155/2020/8864533

Research Article

Wireless Communications and Mobile Computing Blockchain-Based Trust Management in Distributed Internet of Things

Fengyin Li,[1] **Dongfeng Wang**,[1] **Yilei Wang**,[1] **Xiaomei Yu**,[2] **Nan Wu**,[3] **Jiguo Yu**,[4,5] **and Huiyu Zhou**[6]

[1]*School of Computer Science, Qufu Normal University, Rizhao 276826, China*
[2]*School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China*
[3]*Science and Technology Department, Qufu Normal University, Qufu 273165, China*
[4]*School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China*
[5]*Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250014, China*
[6]*School of Informatics, University of Leicester, Leicester LE1 7RH, UK*

Correspondence should be addressed to Yilei Wang; wang_yilei2019@qfnu.edu.cn

Received 10 September 2020; Revised 18 November 2020; Accepted 1 December 2020; Published 19 December 2020

Academic Editor: Shaohua Wan

The development of Internet of Things (IoT) and Mobile Edge Computing (MEC) has led to close cooperation between electronic devices. It requires strong reliability and trustworthiness of the devices involved in the communication. However, current trust mechanisms have the following issues: (1) heavily relying on a trusted third party, which may incur severe security issues if it is corrupted, and (2) malicious evaluations on the involved devices which may bias the trustrank of the devices. By introducing the concepts of risk management and blockchain into the trust mechanism, we here propose a blockchain-based trust mechanism for distributed IoT devices in this paper. In the proposed trust mechanism, trustrank is quantified by normative trust and risk measures, and a new storage structure is designed for the domain administration manager to identify and delete the malicious evaluations of the devices. Evidence shows that the proposed trust mechanism can ensure data sharing and integrity, in addition to its resistance against malicious attacks to the IoT devices.

1. Introduction

Mobile Edge Computing is a new technology which provides an IT service environment and cloud-computing capabilities at the edge of the mobile network. In recent years, with the widespread implementation of the Internet of Things, the number of edge services running on mobile devices has exploded [1]. It is estimated that by 2025, the number of global IoT connections will reach 25.1 billion, and the market size will exceed 10 trillion Chinese yuan. Emerging technologies such as data mining [2], artificial intelligence [3], 5G technology, and natural language processing are also increasingly being extended to IoT applications [4–6]. For example, in the Internet of vehicles [7], we can build a smart city traffic system [8]. Through the use of intelligent abnormal event monitoring for electric vehicles [9] and the use of deep learning to preanalyze road conditions [10], the occurrence of traffic jams can be effectively reduced [11]. Therefore, the need for cooperation between IoT devices has been significantly increased [12]. However, the performance of IOT devices in the process of cooperation is uncertain [13]. The focus of the recent research is how to measure the availability and privacy of data [14, 15] and how to measure the performance of devices through trust data to understand the recent performance of IOT devices [16].

The blockchain technology is essentially a distributed and secure ledger that records all the transactions into a hierarchically expanding chain of blocks. Edge computing brings the cloud capabilities closer to the computation tasks. The convergence of blockchain and edge computing paradigms can overcome the existing security and scalability issues [17]. An IoT device is expected to cooperate with the devices

of high reliability. Before that, it is necessary to ensure the performance of the other devices and the trustworthiness of them, which is the criterion to examine the reliability of the devices before cooperation [18, 19]. However, existing trust mechanisms heavily rely on the trusted third parties or additional trust assumptions; there are hidden security risks such as malicious modifications to the trusted data [20]. Moreover, most distributed trust systems have not considered the malicious evaluation on the IoT devices [21, 22]. Wang et al. proposed a trust management method using environment awareness [23]. From nodes' historical behaviors in different cooperation types, they obtained a comprehensive trustrank to handle any new task, but this process relies on a reliable trust management institution. By caching previous interaction summaries, Liu et al. proposed a verifiable method to solve the hierarchical trust problem of IoT systems [24], but this method needs to establish additional trusted third parties over different domains.

Benkerrou et al. proposed an IoT trust evaluation method based on trust and honesty [25], but they assumed that all master nodes in the domain were completely trusted. Chi et al. proposed an algorithm $SR_{\text{Amplified-LSH}}$ can ensure a good balance between the accuracy and efficiency of recommendation and user privacy information [26]. Based on blockchain technologies, Ren et al. proposed a trust management method suitable for distributed Internet of Things, but they did not consider the irresponsible malicious evaluation problems between malicious devices [27].

Blockchain is a new application of distributed data storage, point-to-point transmission [28], consensus mechanism [29, 30], and encryption algorithms [31, 32]. Blockchain has the characteristics of distributed trust [33], openness, and unforgeability [34], in which the intelligent contract ensures the traceability and irreversibility of transactions. The adoption of multiparty computation and measurement method can guarantee the user to derive results from multiple data sources [35, 36]. Therefore, data sharing and integrity can be guaranteed, and reliable trustworthiness can be established among parties that are blind to each other. Blockchain can realize the sharing and synchronization of trusted data in the distributed Internet of Things, so as to ensure that the data will not be forged or modified by malicious entities [37, 38].

By introducing the theory of blockchain and risk into trust management, we propose a trust management method for distributed IoT. The new mechanism does not rely on any trusted third party; the process of trust establishment and management is entirely independent maintained by each IoT domain manager. The main contributions of our method are as follows:

(1) Aiming at the dependence of trusted third party, a trust mechanism of Internet of Things based on normative trust and risk trust is proposed. This trust mechanism does not rely on any trusted third party, and all trust establishment and trust management are completely managed and maintained by IOT domain administrators and IOT devices

(2) Aiming at the phenomenon of malicious evaluation of devices by existing mechanisms, a trust data storage scheme based on blockchain is proposed. In order to ensure the reliability of the trust mechanism, a storage structure and identification method are designed for domain manager to identify and filter a large number of malicious evaluations of devices

2. Trust Management Model in Distributed Internet of Things

2.1. The Structure of System. According to the characteristics of IoT, we design a decentralized distributed IoT architecture (as shown in Figure 1). Each management domain consists of a domain manager and all subordinate IoT devices. The domain manager manages all IoT devices in the domain. IoT devices can communicate and cooperate with other devices in any management domain. The domain manager can collaborate with others to exchange data.

Each cooperation between the domain manager and the device will be evaluated in both directions based on each other's performance. The gist for evaluation includes the device's communication success rate, data processing capability, transmission range, and network stability. The device can be evaluated based on the other party's overall performance. The communication success rate between the devices is considered as the main indicator of the devices' performance in this paper.

In Figure 1, x represents the IoT domain identifier, x_1, x_2, x_3 represent different IoT domain identifiers, $H(x)$ represents the domain manager of IoT domain x, and $D(x, \mathrm{y}_i)$ represents different IoT devices in the domain x, which is managed by $H(x)$, where $y_i \in N^*(i = 1, 2, \cdots, n)$.

2.2. Trust Model. In order to describe the trustworthiness of IoT devices, this paper uses normative trust and risk measures to quantify trustrank. Normative trust defines the ability of a specific entity to earn credit by other entities, and the risk measure defines the stability level of a specific entity's credit performance in the past period. The concrete definition of the trust model is as follows.

Definition 1. Evaluation value.

The evaluation value of $D(x_i, y_m)$ is denoted as $\delta(x_i, y_m, x_j, y_n, l)$, which refers to the evaluation of a given IoT device $D(x_i, y_m)$ by another IoT device $D(x_j, y_n)$. It is defined as follows.

$$\delta(x_i, y_m, x_j, y_n, l) = \begin{cases} 1, & \text{Good performance,} \\ 0, & \text{Ordinary performance,} \\ -1, & \text{Poor performance,} \end{cases} \quad (1)$$

where l indicates the serial number of the evaluation currently received by $D(x_i, y_m)$.

If the device numbers y_m and y_n are not given here, the evaluation value represents the evaluation value of $H(x_i)$,

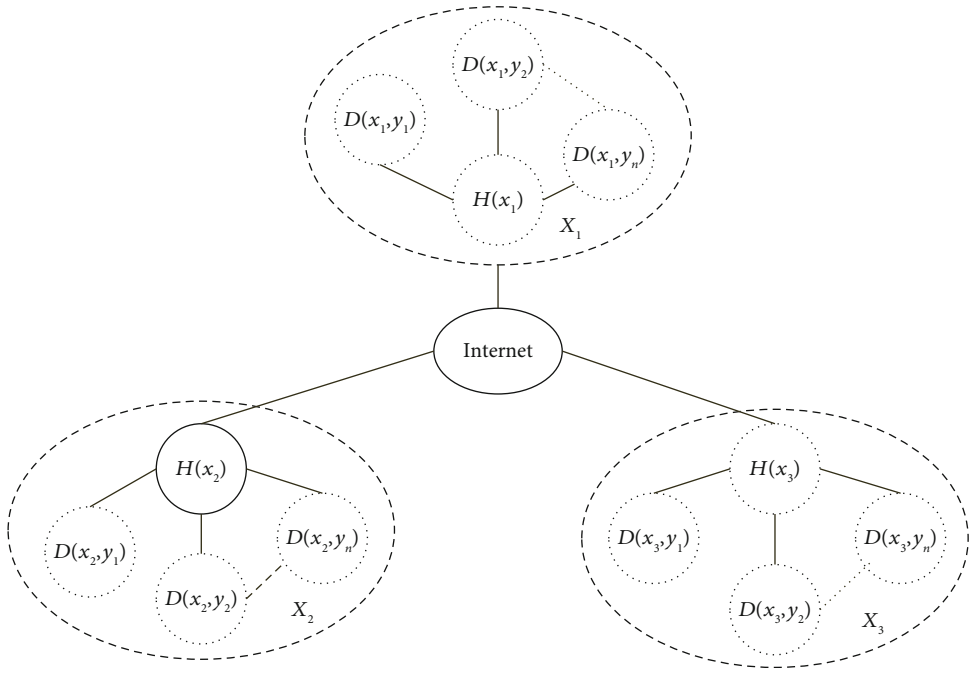


Figure 1: Architecture of distributed IoT.

which refers to the evaluation of a domain manager $H(x_i)$ by another domain manager $H(x_i)$.

Definition 2. Trust scale.

When receiving the kth evaluation, the trust scale of $D(x_i, y_m)$ is denoted as $\mathrm{TC}(x_i, y_m, k)$, and it is iterated according to the evaluated value $\delta(x_i, y_m, x_j, y_n, l)$ given by other evaluators. It is defined as follows.

$$\mathrm{TC}(x_i, y_m, k) = I + \sum_{l=1}^{k-1} \delta\left(x_i, y_m, x_j, y_n, l\right), \tag{2}$$

where I is a trust initial value (we suppose $I = 50$ in our experiments for simplicity) and $k \in N^*$ represents the maximum serial number of the current evaluation received by $D(x_i, y_m)$.

If the device numbers y_m and y_n are not given here, the trust scale represents the trust scale of a domain manager $H(x_i)$, and it is iterated according to its evaluation value given by another domain manager $H(x_i)$.

Definition 3. Normative trustrank.

The normative trustrank of $D(x_i, y_m)$ is denoted as $\mathrm{NT}(x_i, y_m, k)$, which represents the standardized trustrank of device $D(x_i, y_m)$. It is defined as follows.

$$\mathrm{NT}(x_i, y_m, k) = f(\mathrm{TC}(x_i, y_m, k)) = \frac{1}{\left(1 + e^{(-\mathrm{TC}(x_i, y_m, k))}\right)}, \tag{3}$$

where $x_i, x_j (i \neq j)$ represent different IoT domain identifiers, $y_i, y_j (i \neq j)$ represent different IoT devices, and $k \in N^*$ represents the maximum serial number of the current evaluations received by $D(x_i, y_m)$.

If the device numbers y_m and y_n are not given here, the normative trustrank represents the normative trustrank of a domain manager $H(x_i)$.

Definition 4. The mean value.

The mean value of the trust of $D(x_i, y_m)$ is denoted as $\mathrm{MT}(x_i, y_m, k, r)$, which represents the average value of the latest r normative trust of $D(x_i, y_m)$. It is defined as follows.

$$\mathrm{MT}(x_i, y_m, k, r) = \frac{\sum_{k'=k-r+1}^{k} \mathrm{NT}\left(x_i, y_m, k'\right)}{r}, \tag{4}$$

where $k \in N^*$ represents the maximum evaluation serial number received by $H(x_i)$ and $r \in N^*$ represents the number of $CD(x_i, y_m, k')$ included in the risk assessment.

If the device numbers y_m and y_n are not given here, this value represents the mean value of a domain manager $H(x_i)$, which represents the average value of the latest r normative trust of $H(x_i)$.

Definition 5. Risk value.

The risk value of $D(x_i, y_m)$ is denoted as $\mathrm{RV}(x_i, y_m, k, r)$, which is used to measure the risk of the credit performance of $D(x_i, y_m)$ in the history. Up to the maximum evaluation serial number k, the most recent r normative trustranks are taken into consideration, and the risk measure of definition $D(x_i, y_m)$ is as follows.

$$\mathrm{RV}(x_i, y_m, k, r) = \sqrt{\frac{\sum_{k'=k-r+1}^{k}\left[\mathrm{NT}\left(x_i, y_m, k'\right) - \mathrm{MT}(x_i, y_m, k, r)\right]^2}{r}}, \tag{5}$$

where $k \in N^*$ represents the maximum evaluation serial number received by $D(x_i, y_m)$, and $r \in N^*$ represents the number of $\mathrm{NT}(x_i, y_m, k')$ included in the risk assessment.

If the device numbers y_m and y_n are not given here, this value represents the risk value of a domain manager $H(x_i)$, which is used to measure the risk of the credit performance of $H(x_i)$ in the past.

Definition 6. Harmonic trustrank.

The harmonic trustrank of $D(x_i, y_m)$ is denoted as $\mathrm{HT}(x_i, y_m, k, r)$, which is used to represent the comprehensive trust evaluation of $D(x_i, y_m)$. Considering the normative trustrank and risk measure of $D(x_i, y_m)$, we define $\mathrm{HT}(x_i, y_m, k, r)$ as follows.

$$\mathrm{HT}(x_i, y_m, k, r) = \frac{\mathrm{NT}(x_i, y_m, k)}{1 + \mathrm{NT}(x_i, y_m, k) \times \mathrm{RV}(x_i, y_m, k, r)}. \tag{6}$$

If the device numbers $y_m$and y_n are not given here, this value represents the harmonic trustrank of a domain manager $H(x_i)$, which is used to represent the comprehensive trust evaluation of $H(x_i)$.

The architecture of the trust management model is shown in Figure 2.

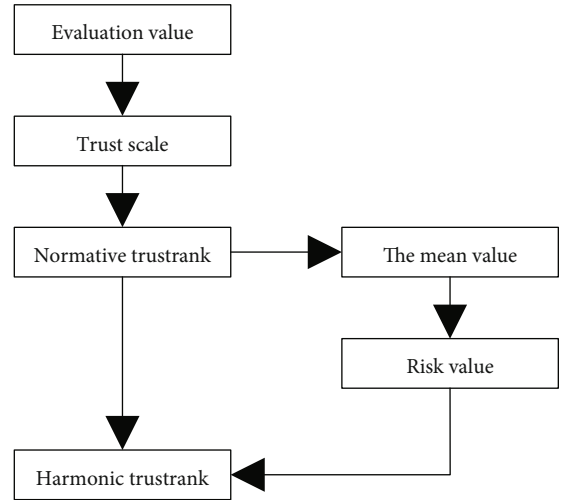


Figure 2: Trust management model.

3. Trust Management Method of Distributed Internet of Things

3.1. Blockchain Structure. In order to achieve trust integrity in data sharing and avoid the existence of irresponsible participants to make a large number of malicious evaluations of other collaborators, a new data structure of the blockchain is designed in this paper, adding the identity of the domain managers, evaluators, evaluatees, and the corresponding evaluation information for providing traceability of the trust evaluation information of the domain managers.

A blockchain can be represented as $\{B_t \mid t \in N^*\}$. $\mathrm{Head}(B_t) \subseteq B_t$ represents the block head, and $B_t - \mathrm{Head}(B_t) = \mathrm{NT}\{\cdot\}$ represents the block body. The trust data in $\mathrm{NT}\{\cdot\}$ is stored in the block body as a Merkle tree, and the root of the Merkle tree is stored in the block head. The block head stores the evaluation information between the domain managers of IoT and the connection information between the blocks. The block body stores the evaluation information between the IoT devices. Taking the evaluation of domain managers $H(x_i)$ and $H(x_j)$ as an example, we define the block structure of the trust data blockchain as follows: $B_t = \{\mathrm{Hash}(B_{t-1}), H(x_i), \mathrm{H}(x_j), \mathrm{TC}(x_j, k-1), \delta(x_j, x_i, k-1), k, \mathrm{MR}, \mathrm{HT}(x_j, k, r), r, \mathrm{NT}\{.\}\}$, where $\sigma_{H(x_j)}(\mathrm{Trl}) = \mathrm{Sig}_{H(x_j)}(\mathrm{Hash}(\mathrm{Trl}))$, $\sigma_{H(x_i)}(\mathrm{Hash}) = \mathrm{Sig}_{H(x_i)}(\mathrm{Hash}(\mathrm{Hash}(B_{t-1}), H(x_i), k, \mathrm{MR}, \mathrm{Trl}, \mathrm{PK}(x_i)))$.

$\mathrm{Hash}(B_{t-1})$ represents the hash value of the previous block B_{t-1}, a block appearing before B_t on the blockchain, $H(x_i)$ represents the identity of the block producer, and $H(x_j)$ represents the identity of the domain manager being evaluated. $\delta(x_j, x_i, k-1)$ is the evaluation value of $H(x_i)$ and $H(x_j)$, which is the $k-1$th evaluation value received by $H(x_j)$. k is the next evaluation's serial number. MR is the Merkle tree root, which is the hash result of the information in the block; Trl is the transaction details of this transaction between $H(x_i)$ and $H(x_j)$. $\sigma_{H(x_j)}(\mathrm{Trl})$ represents $H(x_j)$'s signature on the transaction details Trl. $\mathrm{PK}(x_i)$ is the public key of $H(x_i)$. $\sigma_{H(x_i)}(\mathrm{Hash})$ represents $H(x_i)$'s signature on the transaction information of this block. $\mathrm{MT}(x_j, k, r)$ represents the average value of the latest r normative trust of $H(x_j)$, and $\mathrm{RV}(x_j, k, r)$ represents the trust risk value of $H(x_j)$. $\mathrm{HT}(x_j, k, r)$ represents the harmonious trustrank of $H(x_j)$, and r represents the number of the normative trust included in risk assessment. $\mathrm{NT}\{\cdot\}$ represents the collection of normative trust $NT(x_j, y_n, k)$ of all the other IoT devices $D(x_j, y_n)$ that have been recently evaluated by IoT device $D(x_i, y_m)$. These normative trustranks constitute different records in the IoT domains to which the devices belong, and the domain manager's ID of each IoT domain is marked in the block header.

Assume that there are four sets of device specification trustrank records in the block body, namely, $\mathrm{NT}\{\cdot\} = \{\mathrm{NT}_1, \mathrm{NT}_2, \mathrm{NT}_3, \mathrm{NT}_4\}$, and the structure of the block body is shown in Figure 3.

3.2. Bookkeeping Rights Selection and Block Release. The function of bookkeeping rights selection is to determine which node is used to wrap the trust data, create a block, and then publish it to the blockchain.

3.2.1. Scenario 1. It is a long time for the domain managers $H(x_i)$ and $H(x_j)$ not to cooperate. During this time, it is impossible to share the evaluation results given by the IoT

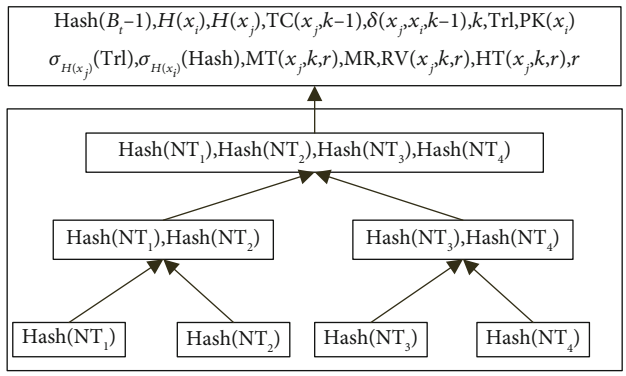


Figure 3: Block structure.

device $D(x_i, y_m)(m = 1, 2, \cdots, n)$ in $H(x_i)$ to the blockchain, and other devices cannot get the latest trust evaluation.

At the moment, firstly, the domain manager $H(x_i)$ detects the utilization rate of the local storage pool. When the utilization rate of the local storage pool reaches a critical value, it performs screening for malicious evaluation of the devices in the IoT domain. If it is determined that there are malicious evaluations of the IoT devices in this domain, these malicious evaluations will be deleted. Then, the normative trustrank of the appraiser is determined by querying the blockchain. The domain manager iterates and updates other valid evaluations of the IoT devices in the domain based on the normative trustrank, generates new blocks, and then publishes them to the blockchain. Meanwhile, the block head uses the fixed format defined in the previous section.

For this purpose, we modify the storage structure of the domain manager so that $H(x_i)$ maintains two fixed-size storage spaces, which are denoted as storage pool Pool1(x_i) and storage pool Pool2(x_i), respectively. Pool1(x_i) is used to receive all the evaluation values $\delta(x_j, y_n, x_i, y_m, l)$ presented by the subordinate equipment $D(x_i, y_m)(m = 1, 2, \cdots, n)$ and sum the evaluation values given by each evaluation device. According to whether or not the sum of the evaluation values exceeds a critical threshold, we can decide whether or not the device has performed a malicious evaluation behavior. If it exists, all the evaluations of the malicious devices will be deleted, thereby blocking all the malicious evaluation data. The evaluation values without any malicious evaluations are assigned to Pool2(x_i).$H(x_i)$ uses the evaluation value $\delta(x_j, y_n, x_i, y_m, l)$ in Pool2(x_i) to obtain the latest trust scale value TC$(x_j, y_n, k')$of $H(x_j)$ using Equation (2) and then uses Equation (3) to calculate the latest normative trust of $H(x_j)$, among which $k' = k + 1$. The detailed operation is as follows.

When the utilization of the storage space of Pool1(x_i) reaches a critical value a, that is, when $a \le ((\text{Occupied}(\text{Pool}1(x_i)))/(\text{Max}(\text{Pool}1(x_i)))) \le 1$ is satisfied, $H(x_i)$ will sum up all the evaluation values $\delta(x_j, y_n, x_i, y_m, l)(n \in N^*)$ of device $D(x_i, y_m)$ in Pool1(x_i) and the sum value is denoted as $S(x_i, y_m)$.

Then, $H(x_i)$ will verify the validity of $S(x_i, y_m)$. If $-\alpha \le S(x_i, y_m) \le \alpha$ is not satisfied, $H(x_i)$ considers this evaluation as malicious and deletes all the evaluations presented by the IoT device $D(x_i, y_m)$.

In the above discussion, Occupied(Pool1(x_i)) represents the storage capacity usage of pool Pool1(x_i), and Max (Pool $1(x_i)$) represents the storage capacity of the pool. a and α are two critical value parameters which represent the storage capacity of the pool Pool1(x_i) and the repeating evaluation times of the evaluators, respectively. Based on the results of our multiple simulations, the performance of the model is well performed while $a = 0.6$ and $\alpha = 6$.

Then, $H(x_i)$ passes the remaining evaluations of Pool1(x_i) to Pool2(x_i). $H(x_i)$ queries on the blockchain $\{B_t \mid t \in N^*\}$ to obtain the latest normative trust NT(x_j, y_n, k) of $D(x_j, y_n)$. According to the deformation of Equation (3), the current trust scale value TC$(x_j, y_n, k) = f^{-1}(\text{NT}(x_j, y_n, k))$ of $D(x_j, y_n)$ can be obtained. If it fails to query the latest normative trust TC(x_j, y_n, k) of $D(x_j, y_n)$ on blockchain $\{B_t \mid t \in N^*\}$, $H(x_i)$ will calculate the trust scale value TC(x_j, y_n, k) of $D(x_j, y_n)$ using Equation (2). Then, using the condition $\text{TC}(x_j, y_n, k') = \text{TC}(x_j, y_n, k) + \sum_{l=1}^{t(x_i)} \delta(x_j, y_n, x_i, y_m, l)$, $H(x_i)$ can calculate the latest trust scale value TC(x_j, y_n, k') of $D(x_j, y_n)$. Finally, $H(x_i)$ calculates the latest normative trust NT(x_j, y_n, k') of $D(x_j, y_n)$ using Equation (3).

In this way, $H(x_i)$ calculates the normative trust NT(x_j, y_n, k') of all the IoT devices $D(x_j, y_n)$ that have been evaluated by other IoT devices in their domain and constructs the block body NT$\{\cdot\}$. All the normative trust NT(x_j, y_n, k') is organized in the form of a Merkle tree where the block head of the MR is added to the new block. $H(x_i)$ will form a new block with the newly generated block head and body NT$\{\cdot\}$ and then publish it to the blockchain.

Since there is no cooperation between $H(x_i)$ and $H(x_j)$ and the latest evaluation value is not obtained, the domain manager related to the cooperation in the block head is set to a specific value ρ (without losing generality, $\rho = 0$). These fields include trust scale value TC$(x_j, k-1)$, evaluation value $\delta(x_j, x_i, k-1)$, serial number k, transaction details Trl, signature $\sigma_{(H(x_j))}$(Trl) of $H(x_j)$on transaction information, mean value of trust MT$(x_j, k\text{-}1, r+1)$, risk value RV$(x_j, k\text{-}1, r+1)$, harmonic trustrank HT$(x_j, k-1, r+1)$, and $r + 1$.

At the same time, domain manager $H(x_j)$ of the domain of all the evaluated devices is stored in Head(B_t), which is convenient for the search of the trust data, and then, the new block $\{\text{Head}(B_t), \text{NT}\{\cdot\}\}$ is released to the blockchain, so as to ensure the timely update of the trust evaluation. The evaluation process is shown in Figure 4.

3.2.2. Scenario 2. When domain managers $H(x_i)$ and $H(x_j)$ cooperate, $H(x_i)$ evaluates the trust of $H(x_j)$ and its subordinate devices after the cooperation. $H(x_i)$ calculates the latest trust data of $H(x_j)$ and each IoT device evaluation result in domain x_j from domain x_i, generates the block head and body of the new block, and then publishes it to the blockchain.

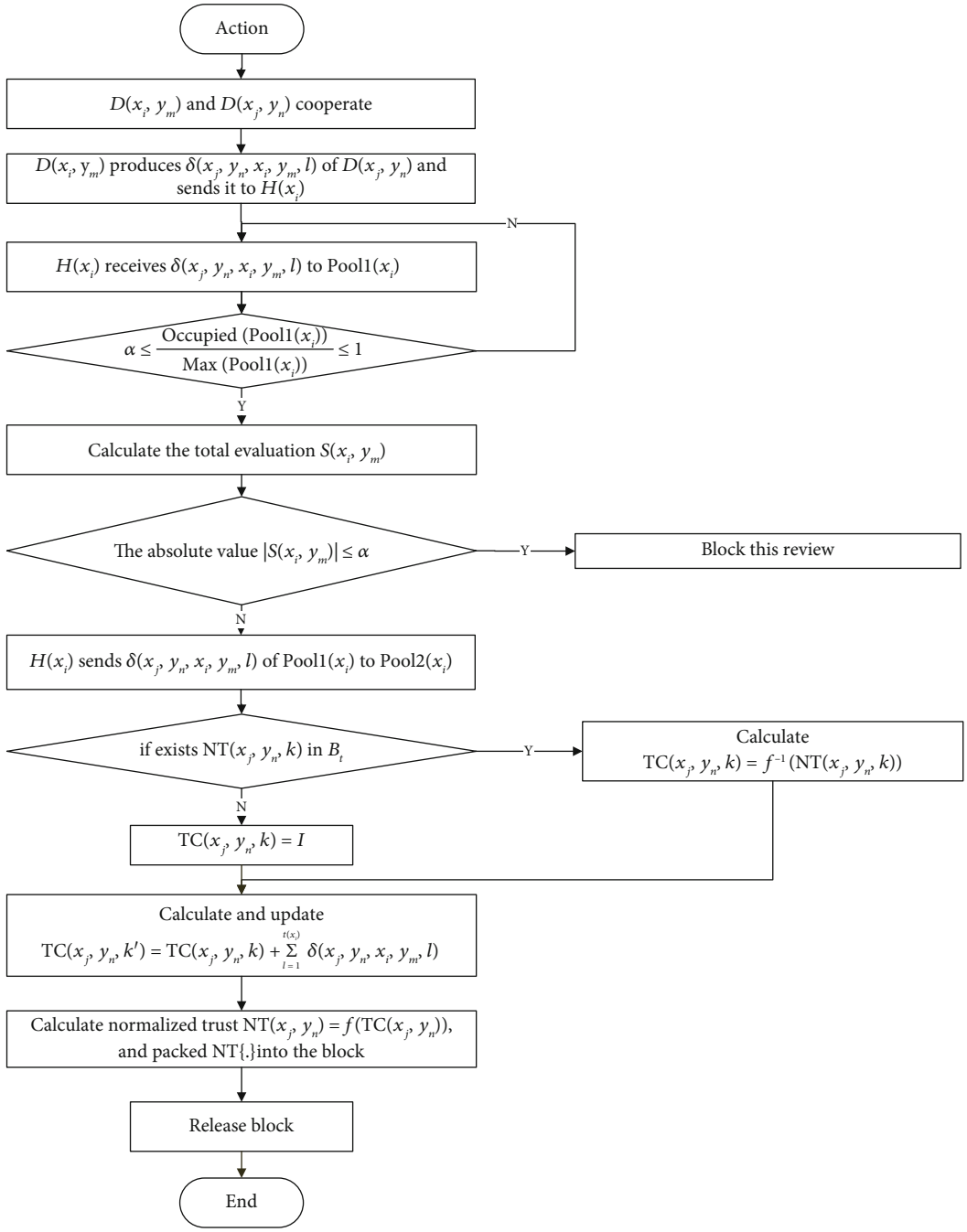
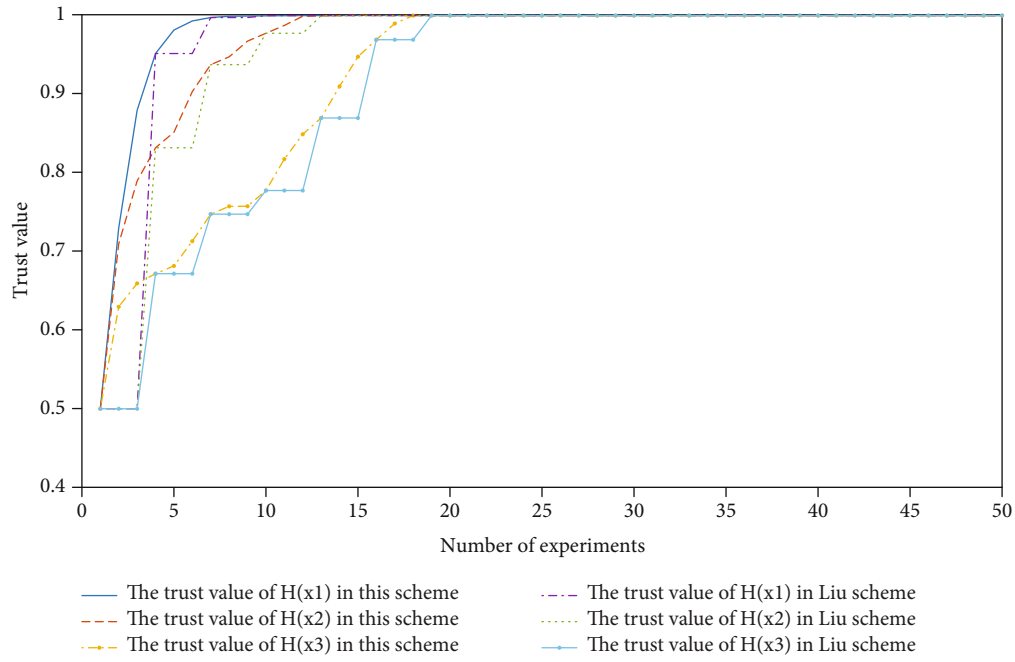


Figure 4: Evaluation flow between IoT devices.

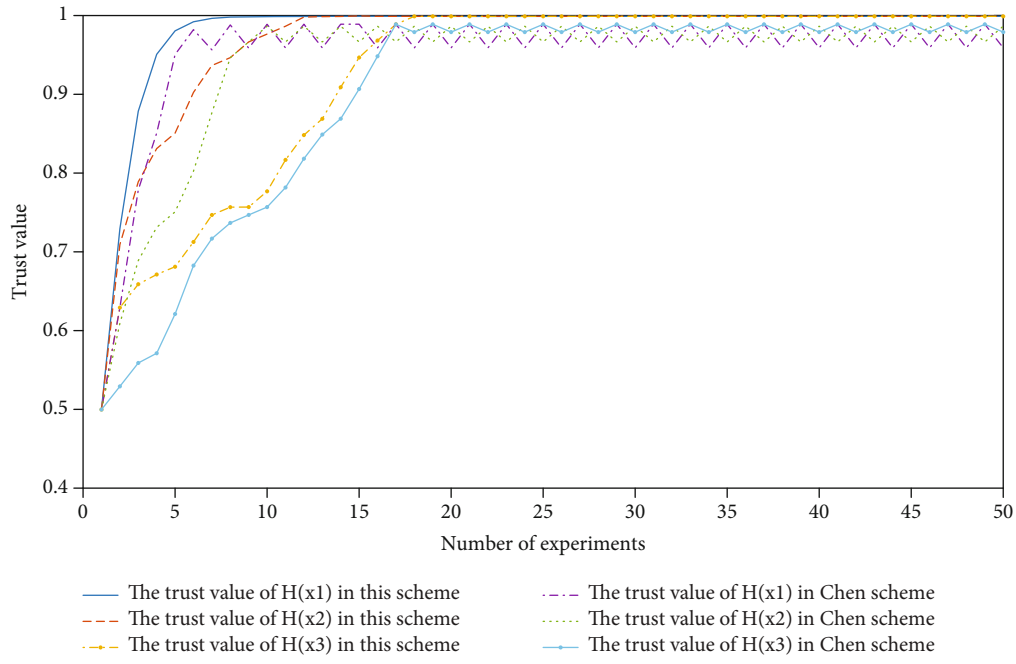
It is assumed that the evaluation value of $H(x_j)$from $H(x_i)$ is $\delta(x_j, x_i, k-1)$ at this time. $H(x_i)$ obtains $H(x_j)$'s current trust scale value TC$(x_j, k-2)$, serial number $k-1$, mean value of trust MT$(x_j, k-2, r)$, risk value RV$(x_j, k-2, r)$, harmonic trustrank HT$(x_j, k-2, r)$, and the number of the included risks r by querying the blockchain. $H(x_i)$ uses Equations (2) and (3) to calculate the latest trustrank TC$(x_j, k-1)$ and the latest normative trust NT$(x_j, k-1)$ of $H(x_j)$.

Then, $H(x_i)$ use Equations (4)–(6) to calculate the mean value of trust MT$(x_j, k-1, r+1)$, risk value RV$(x_j, k-1, r+1)$, and harmonic trustrank HT$(x_j, k-1, r+1)$ of $H(x_j)$. Finally, the fields related to this process are encapsulated in the block head Head(B_t) to form a new block. These fields include $H(x_j)$, TC$(x_j, k-1)$, $\delta(x_j, x_i, k-1)$, k, Trl, $\sigma_{(H(x_j))}$(Trl), RV$(x_j, k-1, r+1)$, HT$(x_j, k-1, r+1)$, and $r+1$.

At the moment, due to the frequent cooperation between domain manager $H(x_i)$ and the other domain managers, the time interval between the two trust data submissions is relatively short. During this period, the number of the evaluation of subordinate IoT device $D(x_i, y_m)(m = 1, 2, \cdots, n)$ stored in storage pool Pool1(x_i) of $H(x_i)$ is relatively small, so it is impossible to judge whether or not these malicious evaluations exist.

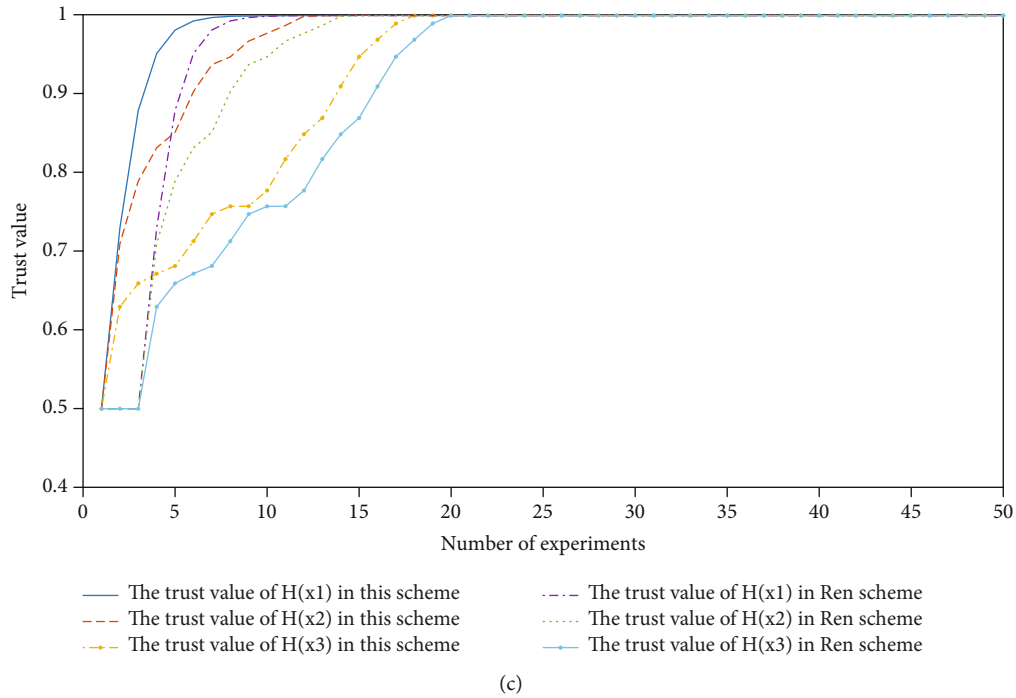


(a)



(b)

Figure 5: Continued.



(c)

Figure 5: Trends of the trustrank of the domain manager. (a) Trend of trustranks of our scheme and Liu's scheme. (b) Trend of trustranks of our scheme and Chen's scheme. (c) Trend of trustranks of our scheme and Ren's scheme.

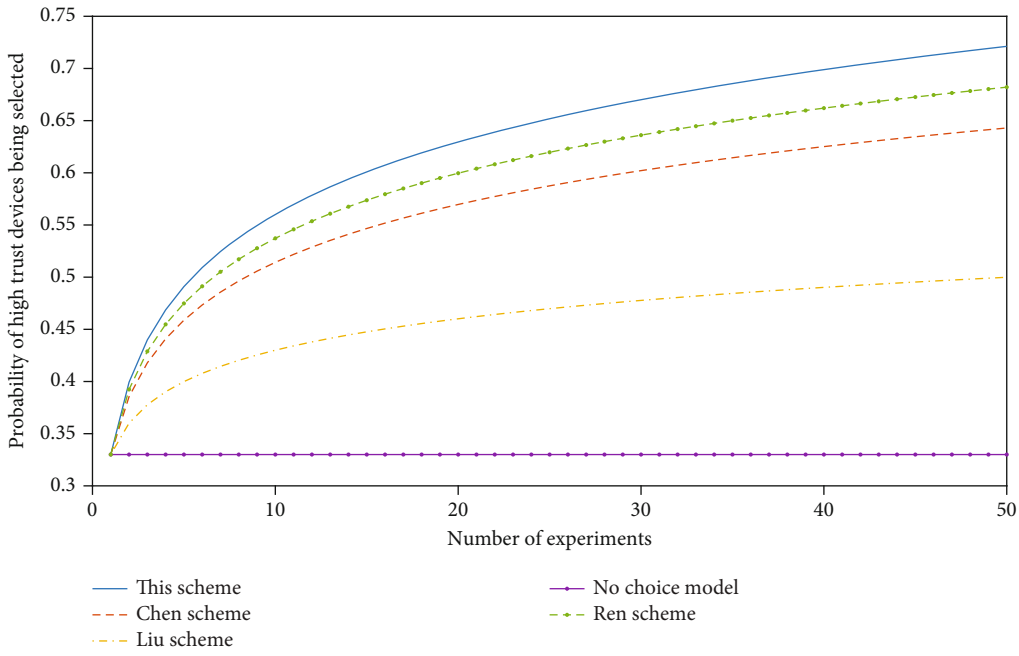


Figure 6: Comparison of the probability of a high-trust device being selected.

In this way, $H(x_i)$ sends all the evaluations stored in Pool1(x_i) to Pool2(x_i). In the subsequent work, $H(x_i)$ will iteratively calculate the evaluation data in Pool2(x_i) to obtain the trust data of the target IoT device and publish it to the blockchain. That is, $H(x_i)$ queries the current normative trustrank NT(x_j, y_n, k) of $D(x_j, y_n)(n \in N^*)$ in the blockchain and calculates the latest normative trust NT(x_j, y_n, k') of $D(x_j, y_n)$ according to Equations (2) and (3), where $k' = k + 1$. Then, $H(x_i)$ generates the block body NT$\{\cdot\}$ from the collection of NT(x_j, y_n, k') and forms a new block together with the block header Head(B_t) formed by the domain manager's trust data and then publish it to the blockchain.

The algorithm for the evaluation between the IoT devices is as follows.

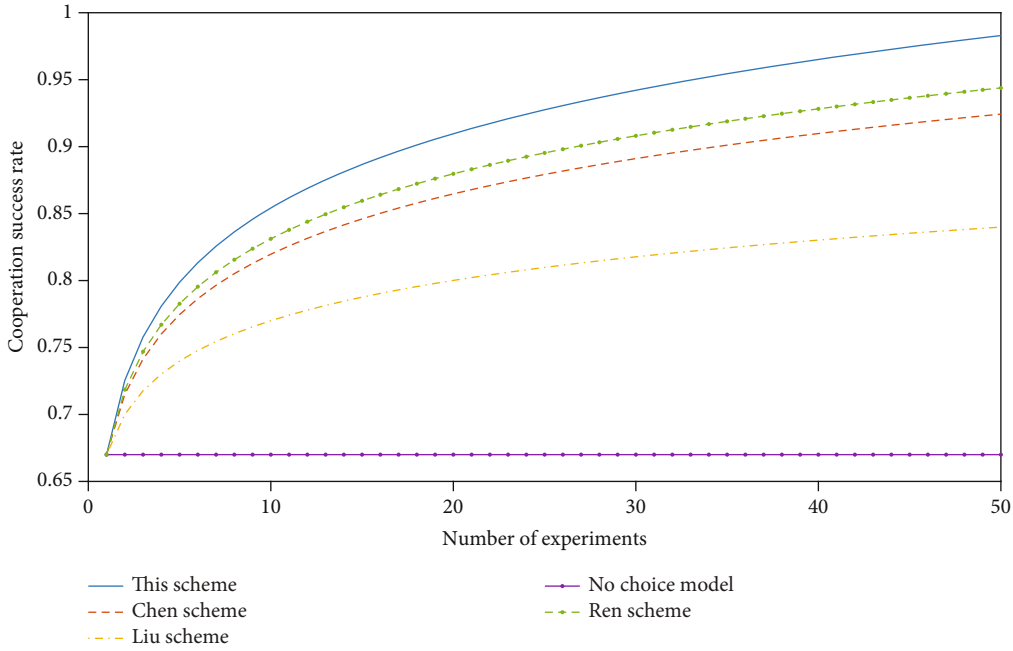


Figure 7: Comparison of cooperation success rates between devices.

Input: evaluation value $\delta(x_j, y_n, x_i, y_m, l)$ $(n \in N^*)$ given by subordinate device $D(x_i, y_m)$ of domain manager $H(x_i)$.

Output: new block B_t.

(1) *Evaluation Collection*. The storage $\text{Pool1}(x_i)$ of the domain manager continuously collects the evaluation value $\delta(x_j, y_n, x_i, y_m, l)$ given by the subordinate equipment and judges whether or not the storage space utilization rate satisfies the inequality $a \le (\text{Occupied}(\text{Pool1}(x_i))/\text{Max}(\text{Pool1}(x_i))) \le 1$ $(a = 0.6)$. If it is satisfied, $H(x_i)$ will sum up all the evaluation values $\delta(x_j, y_n, x_i, y_m, l)$ in $\text{Pool1}(x_i)$ according to the evaluation equipment $D(x_i, y_m)$ to obtain $S(x_i, y_m)$. We then judge whether or not $|S(x_i, y_m)| \le \alpha$ is satisfied. If it is, $H(x_i)$ passes the evaluation value in $\text{Pool1}(x_i)$ to $\text{Pool2}(x_i)$. Otherwise, we delete the evaluation.

(2) *Trust Data Query*. $H(x_i)$ queries on blockchain $\{B_t \mid t \in N^*\}$ to produce the latest normative trust $\text{NT}(x_j, y_n, k)$ of $D(x_j, y_n)$.

(3) *Trust Data Update*. If the query is successful, the current trust scale value $\text{TC}(x_j, y_n, k) = f^{-1}(\text{NT}(x_j, y_n, k))$ of $D(x_j, y_n)$ can be obtained according to the deformation of Equation (6) and then updated according to $\text{TC}(x_j, y_n, k') = \text{TC}(x_j, y_n, k) + \sum_{l=1}^{t(x_i)} \delta(x_j, y_n, x_i, y_m, l)$. If the query fails, $H(x_i)$ calculates the trust scale value $\text{TC}(x_j, y_n, k)$ of $D(x_j, y_n)$ according to Equation (2) and calculates $\text{NT}(x_j, y_n, k)$.

(4) *Block Publish*. The calculated $\text{NT}(x_j, y_n, k)$ constitutes the block body $\text{NT}\{\cdot\}$ of the new block, and $\text{NT}(x_j, y_n, k')$ is organized as a Merkle tree in the block header of the new block. $H(x_i)$ forms a new block together with the block body $\text{NT}\{\cdot\}$ and publishes it to the blockchain.

Table 1: Performance comparison table.

Scheme	Probability of high-trust devices being selected	Cooperation success rate
No choice model	0.33	0.67
Liu's	0.50	0.84
Chen's	0.64	0.92
Ren's	0.68	0.94
Our scheme	0.72	0.97

4. Performance Evaluation

In order to test the effectiveness of the proposed scheme, simulation experiments are carried out to analyze the update rate of trustranks, the probability of the high trustrank equipment being selected, and the success rate of the cooperation.

The experiment simulates three scenarios of the IoT domains and the corresponding IoT devices. The domain manager set is $H = \{H(x_1), H(x_2), H(x_3)\}$, including one malicious device and two benign devices. We used MATLAB to generate evaluation data for 50 device-to-device evaluations, simulating the trend of the trust data in the IoT trust model, the probability of high-trustrank devices being selected, and the success rate of cooperation between IoT devices. All the data are obtained by averaging the results of

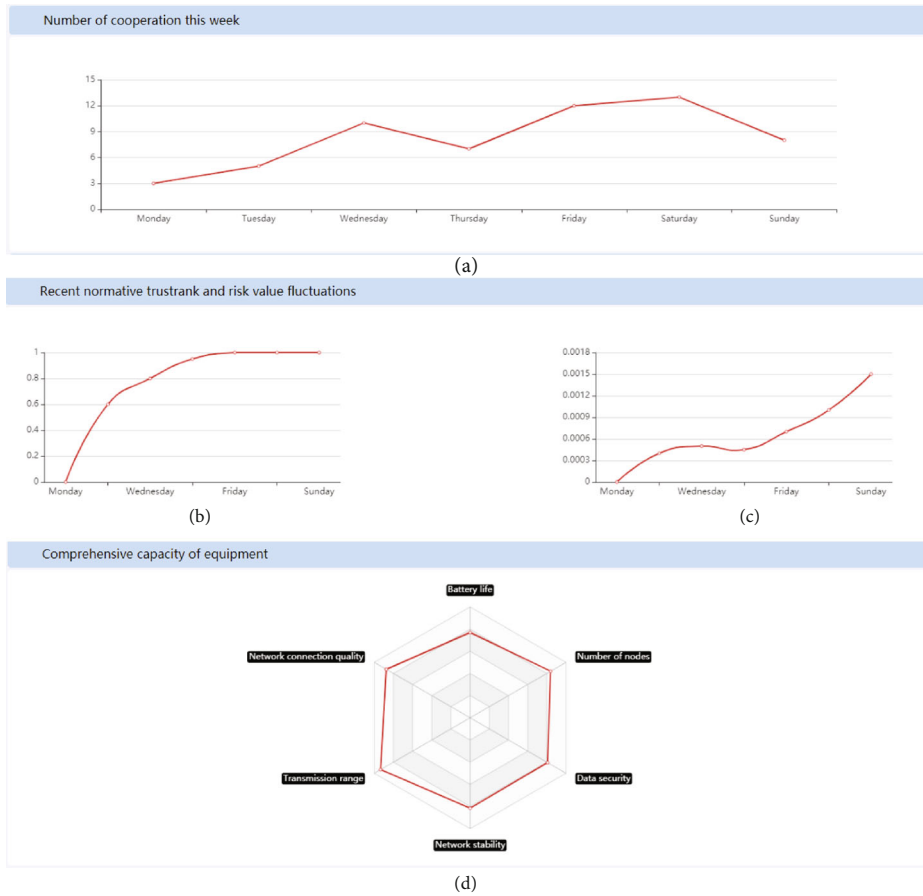


Figure 8: Detailed information of devices. (a) The numbers of cooperation with other devices in the last week. (b) The current trust value. (c) The cooperation stability in the last week. (d) The comprehensive performance analysis.

10 iterations. The experimental results are shown in Figures 5–7.

It can be seen from Figure 5(a) that the trustrank of Liu's scheme [6] is updated every fixed period of time, and the trustrank is not updated in a timely manner. However, the scheme proposed in this paper will immediately update the trustrank after each evaluation by the domain manager, which can reflect the trust status of the IoT devices and the domain managers in a timely manner, and provide more accurate services for the selection of the IoT devices. As can be seen from Figure 5(b), with the increase of the trustrank, in Chen's scheme [19], the trustrank will fluctuate when it converges, and the faster the convergence, the greater the fluctuation of the trustrank, which is not conductive to providing precise services for the selection of IoT devices. However, in this scheme that we proposed, with the increase of the trustrank, when the trustrank converges, the trustrank remains stable and no fluctuation occurs, which is more conducive to providing accurate services for equipment selection. It can be seen from Figure 5(c) that the update of trust in Ren's scheme [11] is slower than the evaluation scheme by almost two evaluation times, so the update speed of trust in this scheme is more timely.

It can be seen from Figure 5 and Table 1 that the selected probability of a high-trust device always remains unchanged at 0.33 in the no-trust model. With the increase of the number of the experiments (the number of the evaluations), compared with Liu's, Chen's, and Ren's schemes, the probability of high-trust devices being selected is steadily increased. However, this scheme has a faster rise rate, and the probability of being selected for high-trust IoT devices is also higher, which can provide a strong guarantee for the subsequent success rate of the cooperation.

It can be seen from Figure 6 and Table 1 that in the trustless model, the cooperation success rate remains unchanged at 0.67 since the IoT device is a randomly selected partner. With the increase of the number of the experiments (the number of the evaluations), the success rate of the cooperation between the devices in this scheme is steadily increasing compared with Liu's, Chen's, and Ren's schemes. However, our scheme has a faster rise rate and a higher cooperation success rate. It can effectively improve the success rate and reliability of the cooperation between the IoT devices.

5. Prototype System

To test the validity of the trust scheme, we implement the system prototype as follows.

5.1. IoT Device Details. The detailed information of the IoT device mainly includes four factors: the numbers of cooperation with other devices in the last week, the cooperation

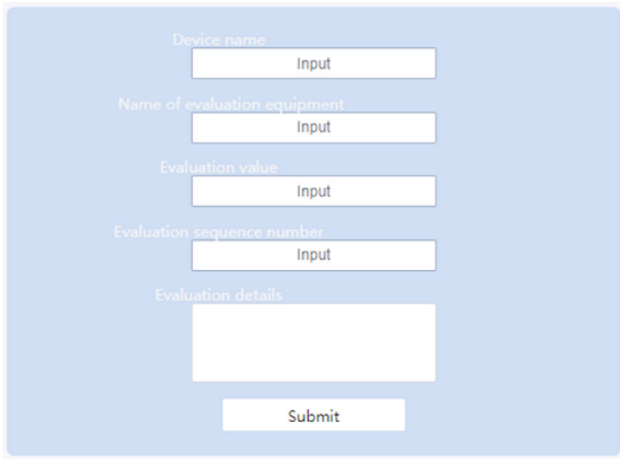


Figure 9: Evaluation operation of IoT devices.

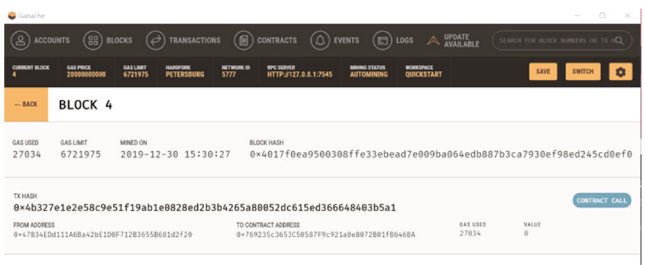


Figure 10: Block detailed information map.

stability in the last week, the current trust value, and the comprehensive performance analysis. Comprehensive consideration of risks can determine the trustworthiness of the device and the expected trust value that can be achieved in cooperation, which can help users select the most trusted devices for cooperation. The gradually changing curves of the detailed information of the IoT device are shown as in Figure 8, showing the four parameters' changing trends.

5.2. Evaluation of IoT Devices. After the users complete the cooperation, they can evaluate the cooperation according to the performance of the other party device. By filling in the information of the evaluated device, the evaluating device, and the evaluation value, the evaluation process is completed by the evaluation submission operation. The evaluation submission model is shown in Figure 9.

The trust management model in the distributed Internet of Things proposed in Section 2 calculates and updates the trust value of the evaluated device, completes the release of blocks by calling smart contracts, and realizes the sharing and synchronization of trust data.

5.3. Trust Data Block Generation. As shown in Figure 10, detailed information such as the block's hash value, block generation address, and contract address is generated. Click CONTRACT CALL to enter the transaction detail information; as shown in Figure 11, we can see the transaction data hash value.

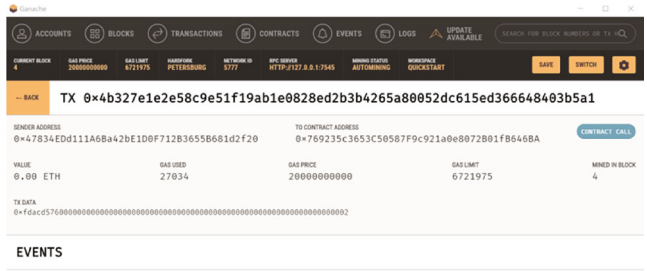


Figure 11: Details of the transactions in the block.

6. Conclusion

Aiming at the problem that the current trust mechanism relies on a trusted third party or additional trust assumptions, which leads to the vulnerability of trust data to malicious attacks, in this paper, we quantify trust into normative trust and risk measure, which can construct a comprehensive review of normative trust, and we propose a trust mechanism for distributed IoT, which modifies the storage structure of the domain manager and realizes the identification and shielding of malicious evaluations between IoT devices, solves the secure storage and sharing of trust data, and can select the device that performs well and stable. Then, it performs well in improving the success rate and reliability of cooperation on IoT devices. However, the mechanism in this paper also increases the storage space requirements of the domain manager, and how to work out this problem is the focus of the future work.

Data Availability

There is no data included in this paper.

Conflicts of Interest

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

Acknowledgments

This work was partly funded by the European Union Horizon 2020 DOMINOES Project (Grant Number 771066).

References

[1] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

[2] X. Yu, H. Wang, X. Zheng, and Y. Wang, "Effective algorithms for vertical mining probabilistic frequent patterns in uncertain mobile environments," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 23, no. 3/4, pp. 137–151, 2016.

[3] X. Zheng and H. Liu, "A scalable coevolutionary multi-objective particle swarm optimizer," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 3, no. 5, pp. 590–600, 2010.

[4] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, p. 106196, 2020.

[5] A. Zhou, S. Wang, S. Wan, and L. Qi, "LMM: latency-aware micro-service mashup in mobile edge computing environment," *Neural Computing and Applications*, vol. 32, no. 19, pp. 15411–15425, 2020.

[6] X. Yu, W. Feng, H. Wang, Q. Chu, and Q. Chen, "An attention mechanism and multi-granularity-based Bi-LSTM model for Chinese Q&A system," *Soft Computing*, vol. 24, no. 8, pp. 5831–5845, 2020.

[7] 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.

[8] C. Chen, Y. Zhang, M. Khosravi, Q. Pei, and S. Wan, "An intelligent platooning algorithm for sustainable transportation systems in smart cities," *IEEE Sensors Journal*, p. 1, 2020.

[9] 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*, vol. 32, no. 9, pp. 4387–4415, 2020.

[10] S. Wan, X. Xu, T. Wang, and Z. Gu, "An intelligent video analysis method for abnormal event detection in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2020.

[11] Y. Cao, H. Song, O. Kaiwartya et al., "Mobile edge computing for big-data-enabled electric vehicle charging," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 150–156, 2018.

[12] J. Wang, "TRM-IoT: a trust management model based on fuzzy reputation for Internet of Things," *Computer Science and Information Systems*, vol. 8, no. 4, pp. 1207–1228, 2011.

[13] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[14] Y. Wang, G. Yang, T. Li, F. Li, Y. Tian, and X. Yu, "Belief and fairness: a secure two-party protocol toward the view of entropy for IoT devices," *Journal of Network and Computer Applications*, vol. 161, p. 102641, 2020.

[15] F. Li, C. Cui, D. Wang et al., "Privacy-aware secure anonymous communication protocol in CPSS cloud computing," *IEEE Access*, vol. 8, pp. 62660–62669, 2020.

[16] Z. Lv, H. Song, P. Basanta-Val, A. Steed, and M. Jo, "Next-generation big data analytics: state of the art, challenges, and future research topics," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1891–1899, 2017.

[17] Y. Wu, H. Dai, and H. Wang, "Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in Industry 4.0," *IEEE Internet of Things Journal*, 2020.

[18] Y. Liu, Y. Gong, and Y. Feng, "Trust system based on node behavior detection in Internet of Things," *Journal of Communications*, vol. 35, no. 5, pp. 8–15, 2014.

[19] X. Li, *Design and Analysis of Revisable Reputation Evaluation System Based on Blockchain [Doctoral Dissertation]*, Xi'an: Xidian University, 2018.

[20] L. Gu, J. Wang, and B. Sun, "Trust management mechanism for Internet of Things," *China Communications*, vol. 11, no. 2, pp. 148–156, 2014.

[21] Y. Ben Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Trust management system design for the Internet of Things: a context-aware and multi-service approach," *Computers & Security*, vol. 39, pp. 351–365, 2013.

[22] X. Liu, X. Yu, H. Zhu, G. Yang, Y. Wang, and X. Yu, "A game-theoretic approach of mixing different qualities of coins," *International Journal of Intelligent Systems*, vol. 35, no. 12, pp. 1899–1911, 2020.

[23] Y. Wang, A. Bracciali, G. Yang, T. Li, and X. Yu, "Adversarial behaviours in mixing coins under incomplete information," *Applied Soft Computing*, vol. 96, p. 106605, 2020.

[24] W. M. LIU, L. H. YIN, B. X. FANG, and H. L. ZHANG, "A hierarchical trust model for the Internet of Things," *Chinese Journal of Computers*, vol. 35, no. 5, pp. 846–855, 2012.

[25] H. Benkerrou, S. Heddad, and M. Omar, "Credit and honesty-based trust assessment for hierarchical collaborative IoT systems," *IEEE*, pp. 295–299, 2016.

[26] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified locality-sensitive hashing-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, 2020.

[27] Y. Ren, X. Li, and H. Liu, "Blockchain-based trust management framework for distributed Internet of Things," *Journal of Computer Research and Development*, vol. 7, pp. 108–124, 2018.

[28] X. Shen, Q. Pen, and X. Liu, "Survey of block chain," *Journal of Network and Information Security*, vol. 11, pp. 11–20, 2016.

[29] D. Li and J. Wei, "Theory, application fields and challenge of the blockchain technology," *Telecommunications Science*, vol. 12, pp. 10–14, 2016.

[30] Y. Wang, A. Bracciali, T. Li, F. Li, X. Cui, and M. Zhao, "Randomness invalidates criminal smart contracts," *Information Sciences*, vol. 477, pp. 291–301, 2019.

[31] S. Wan, M. Li, G. Liu, and C. Wang, "Recent advances in consensus protocols for blockchain: a survey," *Wireless Networks*, vol. 26, no. 8, pp. 5579–5593, 2020.

[32] L. Zhang, Y. Wang, F. Li, Y. Hu, and M. H. Au, "A game-theoretic method based on Q-learning to invalidate criminal smart contracts," *Information Sciences*, vol. 498, pp. 144–153, 2019.

[33] Y. Wang, G. Yang, A. Bracciali et al., "Incentive compatible and anti-compounding of wealth in proof-of-stake," *Information Sciences*, vol. 530, pp. 85–94, 2020.

[34] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, and S. Li, "IPBSM: an optimal bribery selfish mining in the presence of intelligent and pure attackers," *International Journal of Intelligent Systems*, vol. 35, no. 11, pp. 1735–1748, 2020.

[35] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2019.

[36] Y. Wang, M. Zhao, Y. Hu, Y. Gao, and X. Cui, "Secure computation protocols under asymmetric scenarios in enterprise information system," *Enterprise Information Systems on*, pp. 1–21, 2019.

[37] C. Cui, F. Li, T. Li, J. Yu, R. Ge, and H. Liu, "Research on direct anonymous attestation mechanism in enterprise information management," *Enterprise Information Systems*, pp. 1–17, 2019.

[38] Q. Shao, C. Jin, and Z. Zhang, "Blockchain: architecture and research progress," *Chinese Journal of Computers*, vol. 41, pp. 3–22, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8877654, 15 pages
https://doi.org/10.1155/2020/8877654

Research Article

An Automated Real-Time Localization System in Highway and Tunnel Using UWB DL-TDoA Technology

Long Wen,[1] Jinkun Han,[2] Liangliang Song,[3] Qi Zhang,[4] Kai Li,[1] Zhi Li,[4] Weimin Zhang,[5] Beihai Zhang,[5] Xin You,[5] Yunsick Sung,[6] Sumi Ji,[6] and Wei Song[7]

[1]*Beijing Municipal Engineering Research, Beijing, China*
[2]*Department of Computer Science, Georgia State University, Atlanta, 30303 GA, USA*
[3]*Roadway Smart (Beijing) Technology Co., Ltd., Beijing, China*
[4]*Beijing Capital Road Development Group Co., LTD., Beijing, China*
[5]*Beijing Sutong Technology Co., Ltd., Beijing, China*
[6]*Department of Multimedia Engineering, Dongguk University-Seoul, Seoul 04620, Republic of Korea*
[7]*School of Information Science and Technology, North China University of Technology, Beijing 100144, China*

Correspondence should be addressed to Jinkun Han; hjinkun1@student.gsu.edu
and Liangliang Song; liangliang.song@roadwaysmart.com

Received 8 September 2020; Revised 17 October 2020; Accepted 30 October 2020; Published 23 November 2020

Academic Editor: Shaohua Wan

There exists an electromagnetic shielding effect on radio signals in a tunnel, which results in no satellite positioning signal in the tunnel scenario. Moreover, because vehicles always drive at a high speed on the highway, the real-time localization system (RTLS) has a bottleneck in a highway scenario. Thus, the navigation and positioning service in tunnel and highway is an important technology difficulty in the construction of a smart transportation system. In this paper, a new technology combined downlink time difference of arrival (DL-TDoA) is proposed to realize precise and automated RTLS in tunnel and highway scenarios. The DL-TDoA inherits ultra-wideband (UWB) technology to measure the time difference of radio signal propagation between the location tag and four different location base stations, to obtain the distance differences between the location tag and four groups of location base stations. The proposed solution achieves a higher positioning efficiency and positioning capacity to achieve dynamic RTLS. The DL-TDoA technology based on UWB has several advantages in precise positioning and navigation, such as positioning accuracy, security, anti-interference, and power consumption. In the final experiments on both static and dynamic tests, DL-TDoA represents high accuracy and the mean errors of 11.96 cm, 37.11 cm, 50.06 cm, and 87.03 cm in the scenarios of static tests and 30 km/h, 60 km/h, and 80 km/h in dynamic tests, respectively, which satisfy the requirements of RTLS.

1. Introduction

Smart highway integrates and applies advanced information processing technology, sensing technology, and transmission technology [1, 2]. These positioning technologies form an open and common basic platform for monitoring vehicles. Smart highway are aimed at being efficient, convenient, safe, and green, combined with various open operation management and service modes [3], and it provides reliable traffic network service [4, 5] for the rapid transportation of people and goods, provides free communication pipeline service for the vehicle to vehicle [6] or vehicle to road interaction, provides full-time responsive emergency service for emergencies, and provides refined and independent travel service for travelers.

The tunnel project shortens the road distance and reduces the sloping road, so it significantly improves the traffic capacity. Due to the natural electromagnetic shielding effect of the tunnel on radio signals, there is no satellite positioning signal in the tunnel. Therefore, the navigation and positioning service in the tunnel has always been an important problem in the construction of a smart highway. To actively respond to the ideas and development strategy of smart highway construction and improve the industry

localization service, a new solution should be proposed to solve the problem of navigation and positioning service in the highway and tunnel without changing the terminals or increasing the complexity of terminals.

Positioning technology in highway and tunnel is a recognized difficulty in both the industry and academia. The existing positioning technologies mainly contain Bluetooth [7, 8], Wi-Fi [9], Radio Frequency Identification (RFID) [10], ZigBee [11], ultra-wideband (UWB) [12–14], infrared, ultrasonic, etc. Because the traditional UWB technology uses the uplink request mode [15], there is a serious delay problem when six channels are used in the system. This kind of delay problem makes the traditional UWB not suitable for the precise positioning under high-speed driving or the common precise positioning under the action of multiple vehicles.

To achieve the goal of accurate positioning in the high-speed tunnel and solve the problem of navigation and positioning in the tunnel, this paper uses the downlink time difference of arrival (DL-TDoA) technology based on UWB protocol to construct the real-time high-speed positioning system. According to the difference of signal locating time to different monitoring stations, the distance of the signal source can be determined. The advantages of the proposed DL-TDoA system contain no signal-coupling problem, low equipment complexity, and high positioning accuracy. This edge computing system shortens the response time, reduces network pressure, and provides an improved user experience [16, 17].

There is no phase ambiguity in the proposed DL-TDoA system, so the direction-finding baseline cannot be subject to restrictions. The traditional direction-finding method needs to calculate the azimuth angle by the phase, but the phase measurement has the uncertainty of the 2π period. Thus, these traditional methods often use the method that the antenna baseline is smaller than the signal wavelength to avoid the 2π period. However, the wavelength of the high-frequency signal is short, which makes the test antenna close to each other. This way, the high-frequency signal is easy to produce a signal coupling that leads to measurement error. In the proposed DL-TDoA system, each monitoring station only needs one antenna, which fundamentally solves the problem of signal coupling. In each DL-TDoA monitoring station, the system only needs to configure the monitoring antenna and receiver. The requirement of monitoring stations for the antenna is low, even if different monitoring stations use different kinds of antennas. Direction finding antenna is a group of the antenna array, where each antenna in the array is required to keep consistent. Inconsistent antenna arrangement will affect the accuracy of direction finding, which cause high system cost and interfere with monitoring performance. In the DL-TDoA monitoring station, the positioning accuracy depends on the accuracy of time measurement and vehicle speed. To verify the accuracy of DL-TDoA, this paper uses the real-time kinematic (RTK) system to measure the real vehicle trace. Through using a series of comparative experiments, the errors of vehicle speed are displayed in the experiment part. Finally, the system shows that DL-TDoA has high accuracy on the highways and tunnels and the mean errors of 11.96 cm, 37.11 cm, 50.06 cm, and 87.03 cm in the scenarios of static tests and 30 km/h, 60 km/h, and 80 km/h in dynamic tests, respectively, which satisfy the requirements of RTLS.

Specifically, the major contributions include the following:

(1) In the proposed edge computing system, our method not only focuses on the structure of the accurate positioning system but also on the frequency and errors the system requires and the DL-TDoA brings, respectively. Based on the DL-TDoA concept, this paper provides the edge calculating equations that were used in the system and experiment

(2) Some bridge nodes, anchor nodes, and devices for highway positioning and tunnel positioning are introduced. These devices are used in commerce and gain great effects

(3) To evaluate the performance of our approach, we implement all approaches in a true ground, Xishatun test ground. The experimental results show that our approach achieves higher accuracy and efficiency as well as the baseline result of RTK

This paper is organized as follows. In Section 2, some related work is introduced. In Section 3, the proposed DL-TDoA system and how to process data are described. In Section 4, the experiment devices and scenario analysis are presented. We analyze the performance of the experiment results using the proposed DL-TDoA technology and the reference technology, RTK. Section 5 concludes this paper.

2. Related Work

For outdoor real-time positioning, technologies, such as GPS, Beidou [18], inertial navigation, wheel ranging, and ground pseudo base station, have been developed already perfectly. This kind of technology can be combined with a real-time kinematic (RTK) system and can be applied to vehicle navigation and automatic driving in the outdoor environment. However, the research [19] shows that the above-mentioned technology does not apply to the precise positioning and navigation under the high-speed driving conditions in a closed environment such as a tunnel. The cost of some technologies such as inertial navigation is expensive, which is not suitable for the construction and implementation of large-scale projects. To solve the problem of precise positioning in this special case, an indoor real-time high-speed positioning technology is needed.

Bluetooth is a kind of radio technology that supports short-distance communication (generally within 10 m) of equipment. It can exchange wireless information among many devices including mobile phones, wireless headset, and notebook computer. The transmission distance of Bluetooth is 10 cm to 10 m [20]. The Bluetooth connection process involves multiple information transmission and verification processes, repeated data encryption, and decryption process and authentication process for each connection [21]. This process is a great waste and delay for device computing resources, which cannot meet the needs of real-time positioning. Moreover, the security of the encryption algorithm

used in the process of Bluetooth data transmission also needs to be improved.

Radio Frequency Identification (RFID) is a kind of automatic identification technology. It can realize noncontact two-way data communication through radio frequency and read and write the recording media (electronic tag or radio frequency card) by using radiofrequency mode. RFID is used in access control, parking control, production line automation, and material management. RFID technology has some disadvantages: the technology maturity is not enough and the security is not strong enough [22]. RFID technology appeared a short time ago and is not mature in technology. The RFID tag has the characteristics of retroreflection, which makes it difficult to apply in metal, liquid, and other commodities. The security problems faced by RFID technology are mainly manifested in the illegal reading and malicious tampering of RFID electronic tag information.

ZigBee is a wireless network protocol with low speed and short-distance transmission. The main features of ZigBee are low speed, low power consumption, low cost, supporting a large number of network nodes, supporting a variety of network topologies, low complexity, fast, and reliable [23]. The combination of 802.15.4 standard makes its products have the characteristics of low power consumption, easy networking, and short distance interconnection. Therefore, its application in sensor networks or the Internet of things has considerable advantages, but its security and anti-interference performance are lower than those of similar technologies. Despite its excellent performance, ZigBee is also limited by distance.

Infrared communication technology is suitable for low-cost, cross-platform, point-to-point high-speed data connection, especially for embedded systems. Infrared transmission is wireless and cannot transmit too far. There must be no obstacles in the middle, that is to say, it cannot pass through the wall, almost unable to control the progress of information transmission.

Ultra-wideband (UWB) is a new type of wireless communication technology, initially used for military purposes. UWB mainly uses a very short pulse signal to transmit data, which can ensure high-speed communication, but the transmission power is very small [24]. UWB does not need to use the carrier of the traditional communication system but transmits and receives very narrow pulses with nanosecond or below to transmit data, so it has the bandwidth of GHz order. The main advantages of UWB include low power consumption, insensitive to channel fading, strong anti-interference ability, no interference to other equipment in the same environment, strong penetration (positioning in the environment penetrating a brick wall), high security, low system complexity, and accurate positioning. Therefore, UWB technology can be applied to indoor static or moving objects and human positioning, tracking, and navigation and can provide very accurate positioning accuracy.

Download time difference of arrival (DL-TDoA) is a new patented technology of downlink broadcast ultra-wideband. It is a method of positioning by using time difference [25]. The distance of the signal source can be determined by measuring the time when the signal arrives at the monitoring station. The advantage of DL-TDoA is that there is no phase ambiguity, so the direction-finding baseline can be unlimited. DL-TDoA has the characteristics of low complexity, for TDoA monitoring station only needs to configure the monitoring antenna and receiver, and the requirements for the antenna are not high, even if different monitoring points use different antennas. The positioning accuracy of the TDoA detection station depends on the accuracy of time measurement. Through the optimized algorithm, the calculation error of time difference is in the order of 100 ns. Comparing with the other positioning algorithms, like MDS-MAP [26], DL-TDoA outperforms at high-speed calculating.

From the aspect of positioning technology, no matter the positioning accuracy, security, anti-interference, power consumption, etc., the combination technology based on Beidou, DL-TDoA, and visual positioning is one of the most ideal precision positioning and navigation technologies under high-speed driving.

To verify the accuracy of DL-TDoA, an ultrahigh-precision positioning technology is needed for comparison. RTK is a commonly used high-precision GPS measurement and positioning technology. RTK can obtain centimeter-level ultra-precision outdoor positioning results [27]. Through the carrier phase dynamic real-time difference method, the accuracy of positioning and measurement is greatly improved. The RTK method is less affected and limited by visibility, climate, season, and other factors and can implement high-precision positioning in complex terrain areas [28]. Therefore, this paper uses RTK as a comparative reference technology.

3. Automated RTLS Using DL-TDoA Technology

This section describes the proposed automated location solution using a DL-TDoA technology for precise vehicle positioning under highway and tunnel scenarios. The designed architecture of the system is briefly introduced in Sections 3.1–3.3. Section 3.4 introduces the scenario constructions of highway and tunnel. Section 3.5 describes the hardware devices such as tags, anchors, and applications in detail.

3.1. The Comparison with DL-TDoA and Traditional UWB. The main advantages of traditional UWB are low power consumption, insensitive to channel fading, strong anti-interference ability, no interference to other equipment in the same environment, especially when facing lots of vehicles on channel highways, strong penetration (positioning in the environment penetrating a brick wall), and high positioning accuracy. DL-TDoA inherits the excellent features of UWB and develops its new features of real-time positioning. As shown in Figure 1, tradition UBW utilizes the time difference algorithm to calculate the distances between the vehicle and bridge nodes. The UWB router transfers the time differences to the location computing server to calculate the position. DL-TDoA relies on edge computing to reduce the server calculating pressure. In the DL-TDoA framework, the sensor owns edge computing ability to calculate the distances between anchor nodes and the positioning sensor. The sensor transmits the position data to the bridge node with a 4 Hz

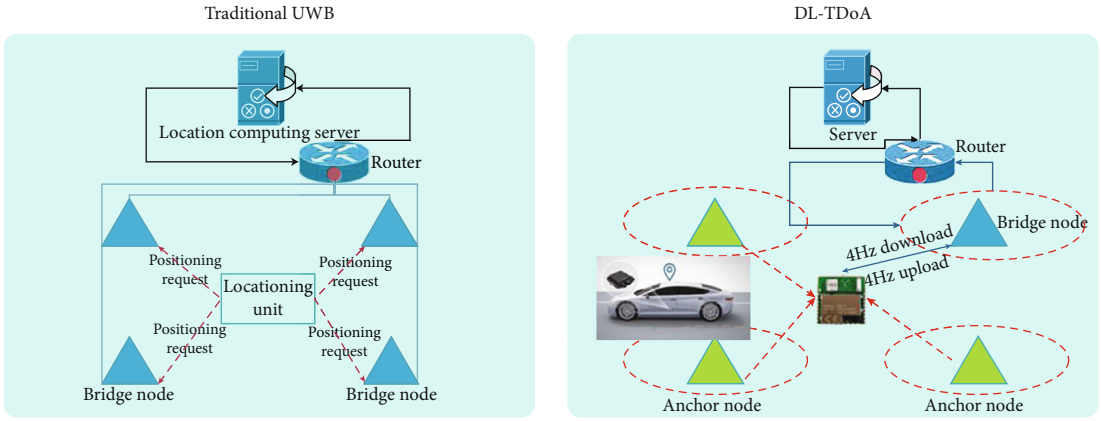


Figure 1: The difference between DL-TDoA and UWB.

upload and download frequency. Then, the router and server just transfer and save the date and the server avoids computing any positioning information. In DL-TDoA, the computing work is assigned to every vehicle to calculate its position, in which way, only one bridge node uploads data while traditional UWB has more than one bridge node. Anchor nodes use the broadcasting protocol, which avoids the time of network verification and save the fee of the public network. Highway and tunnel positioning systems require real-time data so DL-TDoA is better than the traditional UBW to satisfy the real-time requirement and provides a smart mesh network to reduce the computing for the server.

In the traditional UWB system, the data from the positioning tag or sensor is calculated by the server so there is some uploading time. When the car opens the navigation system in the tunnel, the positioning tag has to request the position data from the server so there is some downloading time. The delay of upload and download prevents the traditional UWB system from becoming a real-time positioning system. However, the proposed DL-TDoA uses the edge computing mechanism to obtain a real-time position without any more delay. Moreover, DL-TDoA also uploads the position data to the server for mobile monitoring and analysis. DL-TDoA utilizes the frequency of 1/2/4 Hz to upload, which can be configured by needs, while the USB frequency for DL-TDoA edge computing and navigation can reach 26 Hz without uploading. Traditional UWB cannot serve the navigation applications. Due to the edge computing and the release of server consumption in DL-TDoA, the capacity of the anchor nodes and bridge nodes has no limitations while that of traditional UWB is usually less than 300 vehicles. DL-TDoA uses the anchor nodes and bridge nodes to form the ad hoc network, which utilizes the local network and requires a low price compared with the station using the public network. Therefore, the real-time and low price DL-TDoA solves the problems that the highway and tunnel real-time positioning bring.

3.2. Automated Highway RTLS Overview by Using DL-TDoA. As described in Figure 2, the proposed automated location solution mainly contains 6 groups of devices, including various tags, anchor or bridge nodes, exchangers, routers, networks, and RTLS server. Several common tags are listed in the bottom part of the figure, such as sensor tag, badge tag, asset tag, and a positioning unit. These tags are connected to the wireless infrastructure network (IPV6 mesh), which consists of massive anchor and bridge nodes. The data collected by tags are uploaded to the exchanger through the infrastructure network. When these data continue uploading from the exchanger to the router, their transmission range is converted from a personal domain network to a public domain network. After these data arrive at the RTLS through the network, a series of applications, software, information broad, report, and network services use these data to display locations and analyze results.

According to the mathematical principle, the locus of a moving point whose distance difference from two fixed points is constant is a hyperbola. To determine a point in three-dimensional space, at least three distance differences and four observation points are needed. Therefore, there should be at least four observation stations for positioning.

In the DL-TDoA edge computing tag, the practicability of the localization algorithm affects the result of localization. Assume that the coordinate of the target, a vehicle, is (x, y, z). Now, the DL-TDoA system has at least $M + 1$ anchor nodes, and one of these nodes is the main station S_0 while the others are the substation S_i, and the coordinates are (x_i, y_i, z_i), $i = 0, 1, 2, 3, \cdots, M$. Suppose that the time of electromagnetic radiation from the target to each station is t_i, $i = 0, 1, 2, 3 \cdots M$. The time difference between the arrival times of each substation and that of the main station can be written as π_i, $i = 1, 2, 3, \cdots, M$. By multiplying the time difference of arrival by the speed of light, the distance difference between the vehicle and each substation to the terminal can be obtained:

$$\Delta r_i = c\pi_i, \tag{1}$$

where c represents the theoretical speed of light. The distance difference can also be obtained directly from the

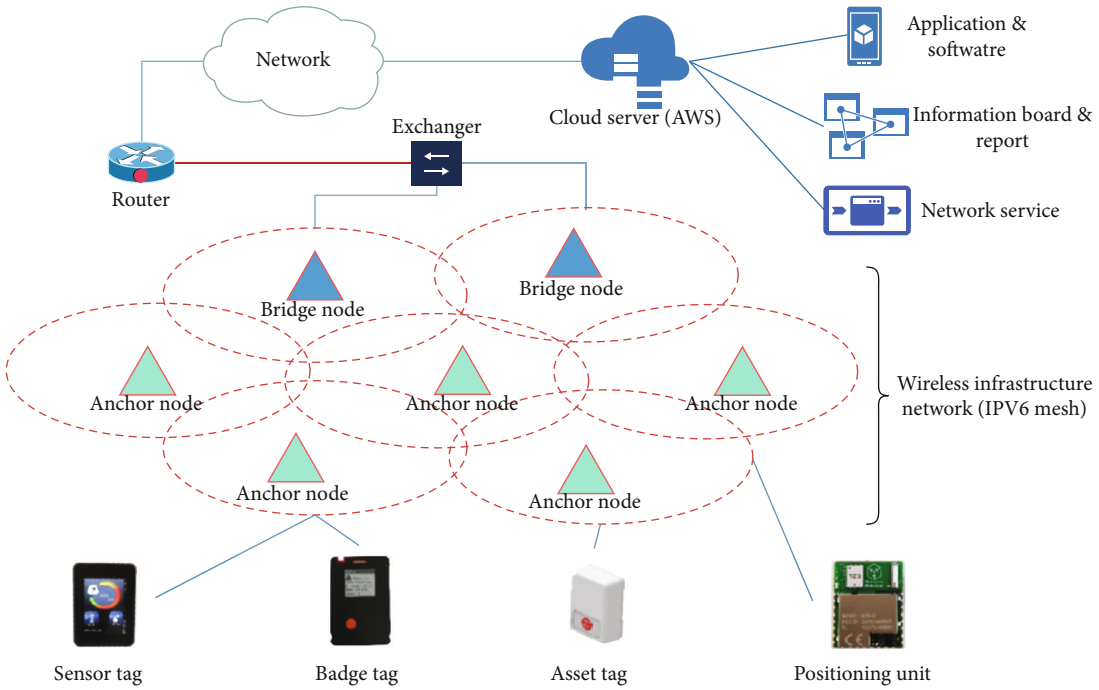


FIGURE 2: The architecture of the proposed automated location solution using DL-TDoA.

distance between the target and the main station minus the distance between the target and the substation:

$$\Delta r_i = r_i - r_0 = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} - \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}. \tag{2}$$

In equation (2), the part of $\sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}$ equals r_0; the next equation is obtained:

$$(\Delta r_i + r_0)^2 = r_i^2 = (x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2. \tag{3}$$

Since r_0 is an unknown parameter, the inequality is reduced to a linear equation and r_0 is eliminated. Subtract r_0^2 from both sides of the formula:

$$\Delta r_i^2 + 2\Delta r_i r_0 = 2[x(x_0 - x_i) + y(y_0 - y_i) + z(z_0 - z_i)] + \left(x_i^2 + y_i^2 + z_i^2\right) - \left(x_0^2 + y_0^2 + z_0^2\right). \tag{4}$$

Let d_i represent the distance of each station, $d_i^2 = (x_i^2 + y_i^2 + z_i^2)$, then

$$\Delta r_0^2 + 2\Delta r_i r_0 = 2[x(x_0 - x_i) + y(y_0 - y_i) + z(z_0 - z_i)] + d_i^2 - d_0^2. \tag{5}$$

It can be sorted out:

$$\Delta r_i r_0 + \frac{\Delta r_i^2 - d_i^2 + d_0^2}{2} = x(x_0 - x_i) + y(y_0 - y_i) + z(z_0 - z_i), \tag{6}$$

where $i = 1, 2, 3, \cdots, m$ of the above equation represents the number of anchor nodes and x, y, z are the unknown numbers, so the above equation is rewritten as the following matrix:

$$\begin{bmatrix} x_0 - x_1 & y_0 - y_1 & z_0 - z_1 \\ \vdots & \vdots & \vdots \\ x_0 - x_m & y_0 - y_m & z_0 - z_m \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \Delta r_1 \\ \vdots \\ \Delta r_m \end{bmatrix} r_0 + \begin{bmatrix} \frac{\Delta r_1^2 - d_1^2 + d_0^2}{2} \\ \vdots \\ \frac{\Delta r_i^2 - d_i^2 + d_0^2}{2} \end{bmatrix}. \tag{7}$$

The matrix of formula (7) can be divided into

$$A = \begin{bmatrix} x_0 - x_1 & y_0 - y_1 & z_0 - z_1 \\ \vdots & \vdots & \vdots \\ x_0 - x_m & y_0 - y_m & z_0 - z_m \end{bmatrix}, \tag{8}$$

$$B = Cr_0 + D = \begin{bmatrix} \Delta r_1 \\ \vdots \\ \Delta r_m \end{bmatrix} r_0 + \begin{bmatrix} \dfrac{\Delta r_1^2 - d_1^2 + d_0^2}{2} \\ \vdots \\ \dfrac{\Delta r_i^2 - d_i^2 + d_0^2}{2} \end{bmatrix}. \tag{9}$$

According to the linear properties of linear equations, the solution set of $AX = B$ is the sum of the solution sets of $AX = Cr_0$ and $AX = D$. When $m = 3$, A is a square matrix. According to the Cramer rule, its solution can be expressed as

$$x_{ij} = \frac{|A_j|}{|A|}, \tag{10}$$

where A_j is the determinant obtained by replacing the jth column element in a with a constant term. The new equation can be obtained:

$$\begin{cases} x = a_1 r_0 + b_1, \\ y = a_2 r_0 + b_2, \\ z = a_3 r_0 + b_3, \end{cases} \tag{11}$$

where a_i is the solution set of $AX = C$ and b_i is the solution set of $AX = D$. Then, replace the x, y, z variables of equation (12) with those in equation (11):

$$r_0^2 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2, \tag{12}$$

$$r_0^2 = \left(a_1^2 + a_2^2 + a_3^2\right) r_0^2 + 2r_0(a_1 b_1 + a_2 b_2 + a_3 b_3 - a_1 x_0 - a_2 y_0 - a_3 z_0) + (x_0 - b_1)^2 + (y_0 - b_2)^2 + (z_0 - b_3)^2. \tag{13}$$

Suppose $\alpha = (a_1^2 + a_2^2 + a_3^2)$, $\beta = (a_1 b_1 + a_2 b_2 + a_3 b_3 - a_1 x_0 - a_2 y_0 - a_3 z_0)$, and $\gamma = (x_0 - b_1)^2 + (y_0 - b_2)^2 + (z_0 - b_3)^2$, then the final equation is

$$r_0 = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}. \tag{14}$$

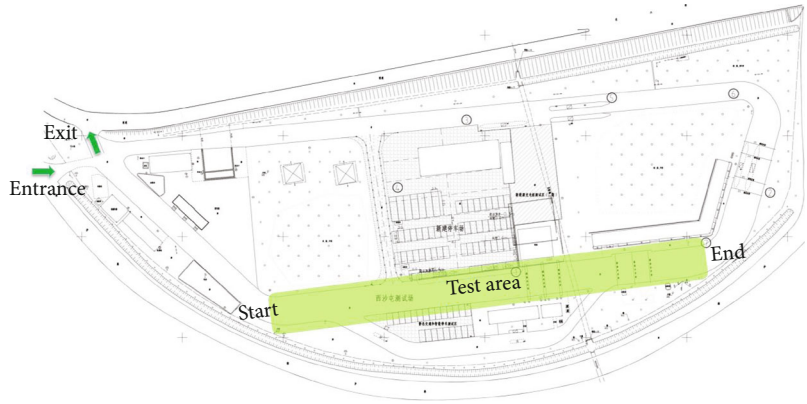
When r_0 has two solutions, two positioning results are obtained, that is, the positioning is fuzzy, so the observation station needs to be added. When r_0 has a solution, the target position can be uniquely determined. When r_0 has no solution, the position cannot be determined. After r_0 is obtained, x, y, z can be found in equation (11). Through the edge calculation of the positioning tag on the vehicle, the position information returned by the DL-TDoA algorithm is calculated in the tag to carry out accurate positioning processing.

3.3. RTK and DL-TDoA Data Format Processing. RTK positioning system (high-precision GPS measurement and positioning technology), as a comparative technology, DL-TDoA data should be transformed into RTK format in the same coordinate. The original longitude and latitude of RTK data are transformed as international format: degrees, minutes, and seconds. Since the original RTK data collection is based on the geodetic coordinate system, the WGS-84 ellipsoid datum algorithm transforms the coordinate system of the data from the geodetic coordinate system to a x, y, z plane coordinate system. In this case, the origin of the coordinate system is not the origin specified by the test site, so it is necessary to translate the data to the coordinate system centered on the specified origin. The data on the other side is DL-TDoA data. The data of DL-TDoA is the data of the plane coordinate system, but the data needs to be rotated to the same x, y, z direction the same as RTK data. Since the uplink DL-TDoA frequency is 4 Hz, while the RTK upload frequency is 26 Hz, it is necessary to use the difference compensation method to find the missing values, to achieve the same frequency as RTK. As the RTK and DL-TDoA antenna cannot be fixed at the same position on the vehicle, there is a physical error to compare the traces. Therefore, the data from DL-TDoA should manually minus the error distance. Now, the RTK and DL-TDoA data are in the same coordinate, and no outside errors do not come from algorithms.

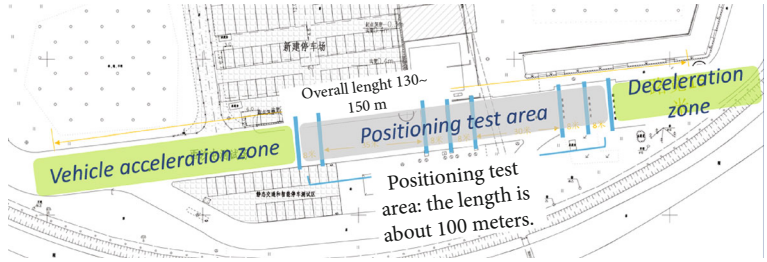
3.4. Highway and Tunnel Scenario Setup. As shown in Figure 3(a), the highway simulation environment is selected in the Xishatun test ground. The green area is the test area, and the two ends are the start point and the endpoint, respectively. At this time, the area is a straight barrier-free road, and there is no signal occlusion. The test area is described (Figure 3(b)). First of all, the acceleration part and deceleration part of the vehicle are green areas, and the green area is not included in the data analysis plan, because the main test of DL-TDoA is still focused on driving at a constant speed. The total length of the test area is 130-150 m, and the length of the DL-TDoA positioning test area is 100 m. Nodes are set in the gantry area as shown in the test (Figure 3(c)). Due to the natural conditions of the test area, the distance between the gantries is not the same. The interval is about 8 m or 30 m. To increase the positioning effect, the test area requires some artificial anchor points, which may not be fixed on the gantry because the number of gantries is fixed. Therefore, temporary anchor points for tripod support are added on both sides of the road in Figure 3(d) (green anchor points are temporary anchor points while red ones are fixed anchor points fixed on gantry). Anchors 7 and 8 are bridge nodes, which link to the public network to upload the positioning track. The accuracy of DL-TDoA on the highway is verified to determine whether it can perform well in the tunnel.

The arrangement of anchor nodes and bridge nodes on both sides of the tunnel is shown in Figure 4. Because there are rocks and walls on both sides of the tunnel, it is more effective to arrange joints on the wall. Anchor nodes and bridge nodes are arranged alternately to obtain a better ad hoc network.

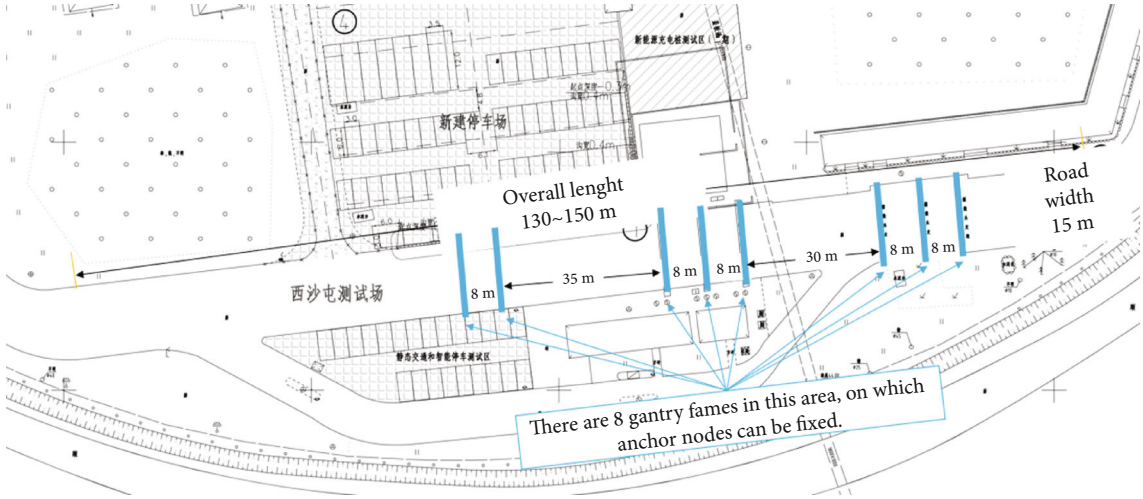
3.5. Anchor and Tag Devices. To solve the time-consuming and labor-consuming problem of coordinate measurement in the anchor node, the self-positioning module of the anchor node is developed in this paper. After the installation of the



(a)

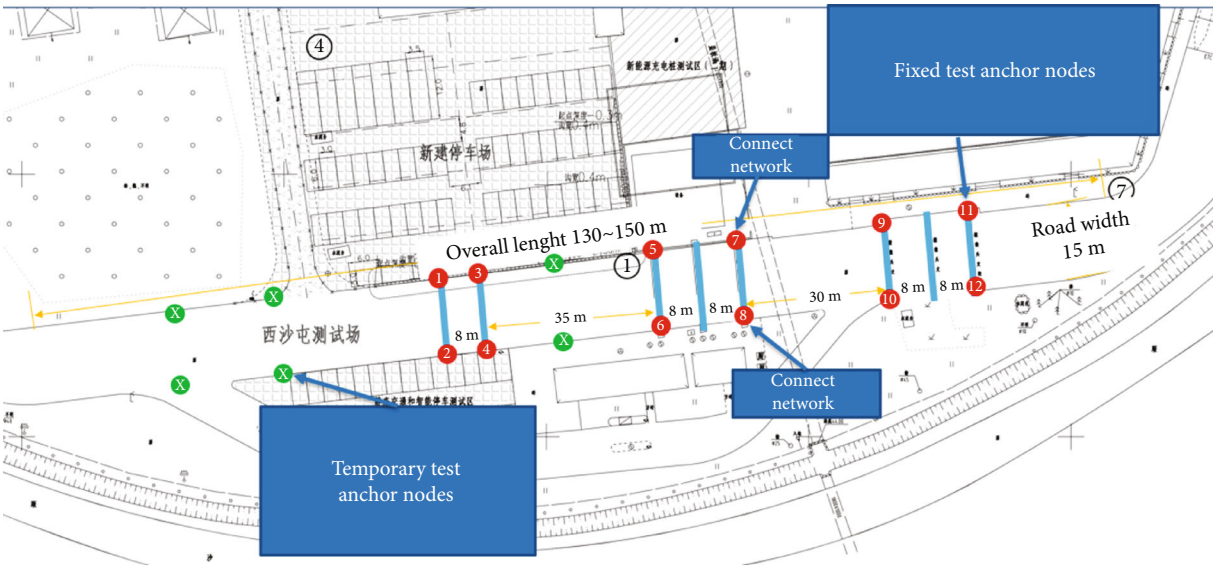


(b)



(c)

Figure 3: Continued.



(d)

Figure 3: Anchor and bridge node deployment for the DL-TDoA highway test. (a) Test area: Xishatun test ground. (b) The test area includes three parts: the acceleration zone, the test area, and the deceleration zone. (c) Gantry distances. (d) The distribution of anchor and bridge nodes.

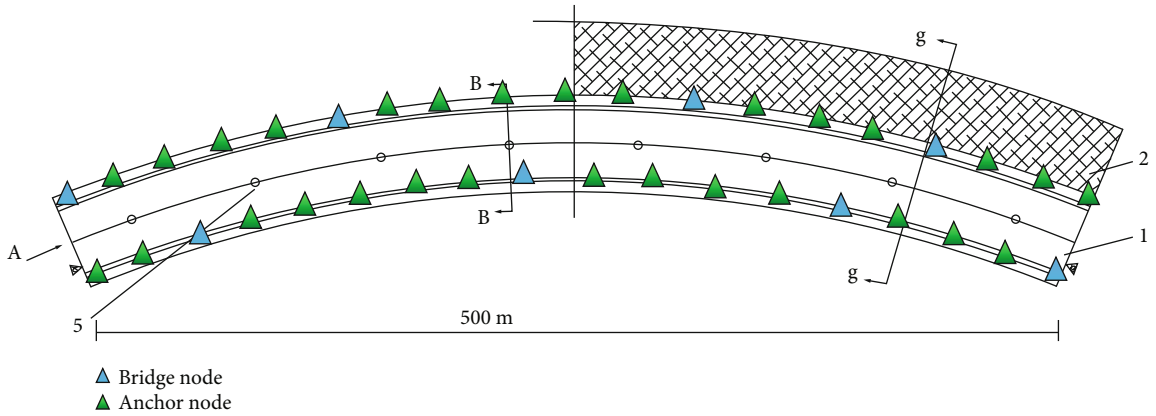


Figure 4: The distribution of anchor and bridge nodes in the tunnel test.

(a) (b) (c)

Figure 5: Three common anchor node devices adopted in the proposed solution. (a) Wall-mounted anchor node. (b) Roof anchor node. (b) Three-defense anchor node.

equipment, the system can automatically measure the coordinates of anchor nodes, which establish a coordinate system and simplify the construction process. Figure 5 displays three common anchor node devices, including wall-mounted anchor nodes, roof anchor nodes, and three defense anchored nodes. The wall-mounted anchor node in Figure 5(a) can be hanging fixed on the vertical wall. The roof anchor node in Figure 5(b) is always fixed with clamps or screws on indoor ceilings. The three-defense anchor node in Figure 5(c) adopts a screw fixation method that can be installed in both indoor and outdoor environments. All three kinds of anchor nodes are powered by PoE supply.

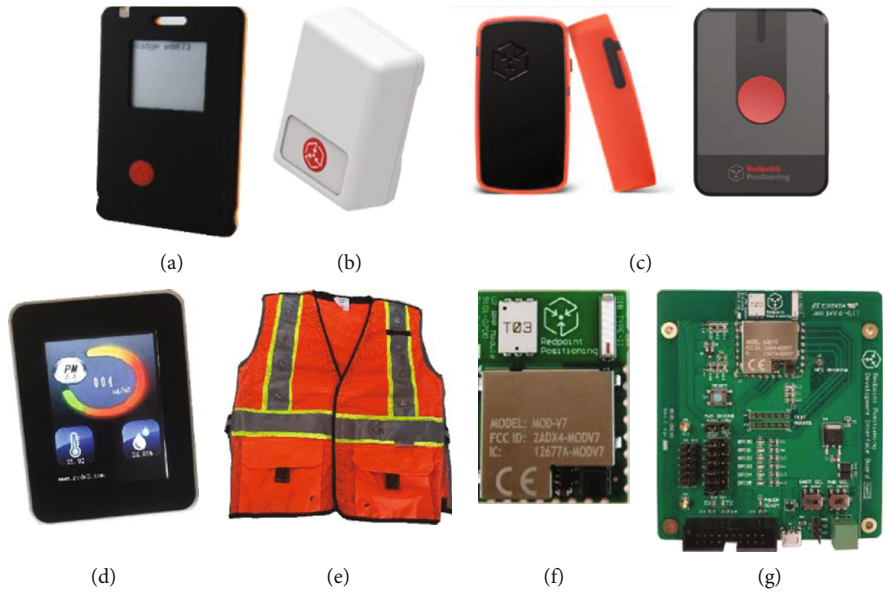


Figure 6: Seven common tags and units adopted in the proposed solution. (a) Badge tag. (b) Asset tag. (c) General tag. (d) Sensor tag. (e) Safety vest. (f) Positioning unit. (g) V7 development board.

In the proposed solution, the wireless network connects the anchor nodes, where each anchor node only needs PoE power. About 15% of anchor nodes in the network need to be connected to the switch. In the construction process of the proposed solution, all wireless networks make the deployment of anchor nodes easier and the construction cost is greatly reduced. Based on the calculation of the same number of base stations, the cost of construction and auxiliary materials of our solution will be saved by about 70%. Moreover, because most anchor nodes need power supply tightly, they will be more flexible in the location selection of base stations.

The DL-TDoA algorithm proposed in this paper can use the tag to calculate the location information, making the capacity of the network theoretically unlimited. In the actual application scenario, it is enough to support a large number of tags. Based on wireless communication technology, anchor nodes automatically form an IPv6 mesh network, which has very flexible scalability. If there exists a deployed base station under the environment, the newly installed base station will automatically join the existing mesh network if the location area needs to be increased.

Figure 6 displays seven common tags and positioning units, including badge tag, asset tag, general tag, sensor tag, safety vest, positioning unit, and V7 development board. In Figure 6(a), the badge tag is always used for tracking personnel. The device owns one button with an alarm function. When the alarm is triggered, the device makes a sound, its LED stroboscopic alarm, and its electronic screen display. The device is wearable and with a single rechargeable lithium battery. Figure 6(b) displays an asset tag that is always used to track assets or objects with low moving frequency. The device can be fixed on objects, which also contains one alarm button and an LED stroboscopic alarm. The general tag in Figure 6(c) is always used to track vehicles or people, which can be fixed to objects or carried by people. The device owns an LED stroboscopic alarm with a sensor interface. The sensor tag in Figure 6(d) integrates a positioning module that can collect and upload sensor data with time and space labels. The device supports integration in the third party to design personalized products. The safety vest as shown in Figure 6(e) owns a wearable positioning label with light prompt and alarm function. It is suitable for the production environment of various industries: lithium battery (rechargeable) power supply. The positioning unit in Figure 6(f) contains complete core positioning functions, the firmware is preloaded, and the development of various application interfaces is supported. Support the design of personalized labels or other products on the third-party PCB. The PCB of the V7 development board in Figure 6(g) integrates a positioning module that provides multiple interfaces for the third party. Its external power supply is required as no more than 3.6 V through the USB power supply.

Table 1: Device list used in test.

No.	Device name	Count	No.	Device name	Count
1	Bridge node	2	7	Router	1
2	Anchor node	16	8	Distance meter	2
3	Location server	1	9	Total station	1
4	PoE exchanger	1	10	Laptop	2
5	Network cable	200 m	11	PoE extender	2
6	V7 general tag	4			

Each tag has its computing power and can calculate the location information independently. Considering the real-time requirement in RTLS, our tags can compute the location information by themselves, to achieve almost zero delays. For scenes with high real-time requirements, such as anticollision and unmanned driving, the proposed system has great advantages. The proposed solution has realized the active control of

(a) (b) (c) (d)

Figure 7: Scenario was set up. (a) Installing anchor nodes. (b) Fixed nodes. (c) Testing in different positions. (d) Collecting data.

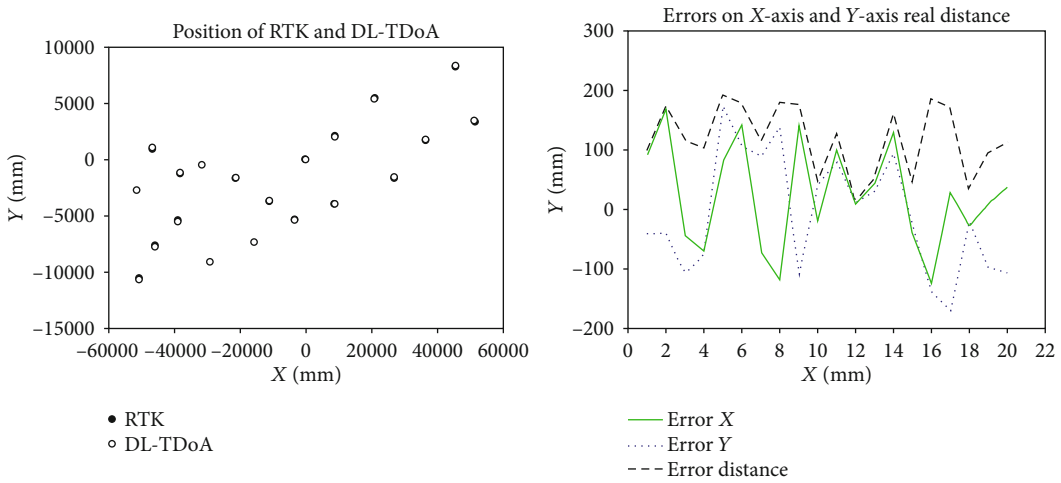


Figure 8: Static DL-TDoA and RTK test comparison.

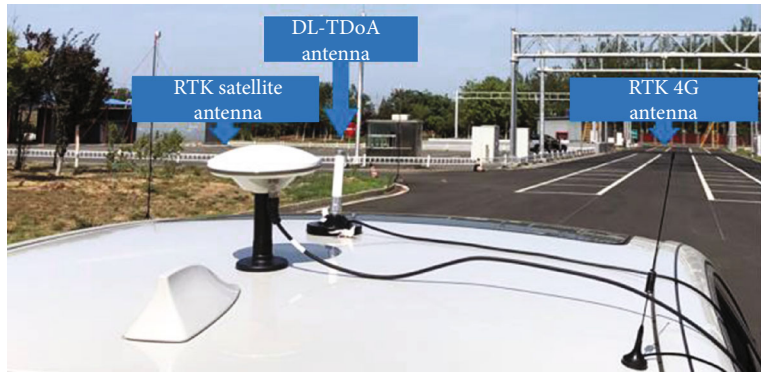
a forklift and other machinery in the storage field and has made a lot of technical reserves for unmanned driving.

4. Experiments and Scenario Analysis

The verification experiment of this paper is divided into two application scenarios: highway and tunnel internal test. The devices used in the experiment are listed in Table 1. We use 2 bridge nodes, 16 anchor nodes, 1 location server, 1 PoE exchanger, 4 V7 general tags, 1 router, 2 distance meters, 1 total station, 2 laptops, and 2 PoE extenders. Moreover, a 200 m network cable is used in the experiment.

4.1. Scenario 1: Highway Static Test. Before the test, the devices and sensors are fixed on gantries and connected to the IPV6 network and local network. As shown in Figures 7(a) and 7(b), the anchor and bridge nodes are fixed on gantries. To verify that the proposed DL-TDoA is accurate enough, first of all, the static position comparative experiments are needed. Testers prepare a triangle bracket (Figure 7(c)), where the RTK positioning sensor and DL-TDoA positioning sensor are alternated. The fixed position of the triangle bracket guarantees the same position for both two positioning sensors so there are no sensor physical position errors. Testers choose several positions to collect the position data for future comparisons. Figure 7(d) shows that the positioning sensor connects the laptop directly in the static test scenario. Static test data is collected by local USB transmission.

The RTK data obtained after the fixed-point test will be converted into a plane coordinate system (using WGS-84 ellipsoid datum). It is easy to transform the RTK from the origin of the plane coordinate system to the coordinate system with the fixed point of the test field as the origin. For the data of DL-TDoA, the plane coordinate system is used as the north direction of the y-axis, and the original coordinate system adopted by DL-TDoA is rotated. Because the sensor positions of RTK and DL-TDoA are the same, the position difference between RTK and DL-TDoA antenna is removed and the coordinates are corrected. Finally, the distance difference between RTK and DL-TDoA is calculated point to point, and the average value of the distance difference is calculated



(a)

(b)

(c)

Figure 9: Dynamic scenario was set up.

to get the final result. Figure 8(a) shows the RTK and DL-TDoA data in the same rectangular coordinate system. Black points represent RTK while the white points present the DL-TDoA data. As shown in the figure, the majority of the tested positions is overlapped and provides excellent accuracy. For some point, only the DL-TDoA data can be seen, which proves RTK and DL-TDoA data is full the same under the millimeter unit. Figure 8(b) shows the error under static positioning. It is seen from Figure 8(b) that the maximum error is less than 200 mm and the minimum error is close to 0 mm. The maximum distance error is 193.12 mm, and the minimum distance error is 14.43 mm. The average error of RTK and DL-TDoA was 119.64 mm.

4.2. Scenario 1: Highway Dynamic Test. For the dynamic test, RTK and DL-TDoA sensors are fixed on the top of the vehicle. As shown in Figure 9(a), there is a 20 cm distance between two kinds of antennas, which causes the deviation when tracing the trajectory of two motions. Therefore, comparing with the static test, when processing the data, testers are required to manually modify the deviation, by simply subtracting 20 cm. Another problem is that RTK transmits the data with a 26 Hz frequency while the frequency of DL-TDoA is only 4 Hz. Tester manually finds the same position in the same frame. These two problems are the main differences between the static and dynamic tests. Figures 9(b) and 9(c) are the start and end positions for the dynamic test. Then, the vehicle goes through the gantries, which are fixed with DL-TDoA anchor nodes and bridge nodes. The overall distance of the dynamic test is 100 m.

In the dynamic test, three comparative experiments are set to speeds of 30, 60, and 80 km/h, respectively. For the dynamic data format, due to the deviation between RTK and DL-TDoA antenna, testers are supposed to minus the distance of that error on the coordinate. Another problem this paper has mentioned is that the uploading frequency of

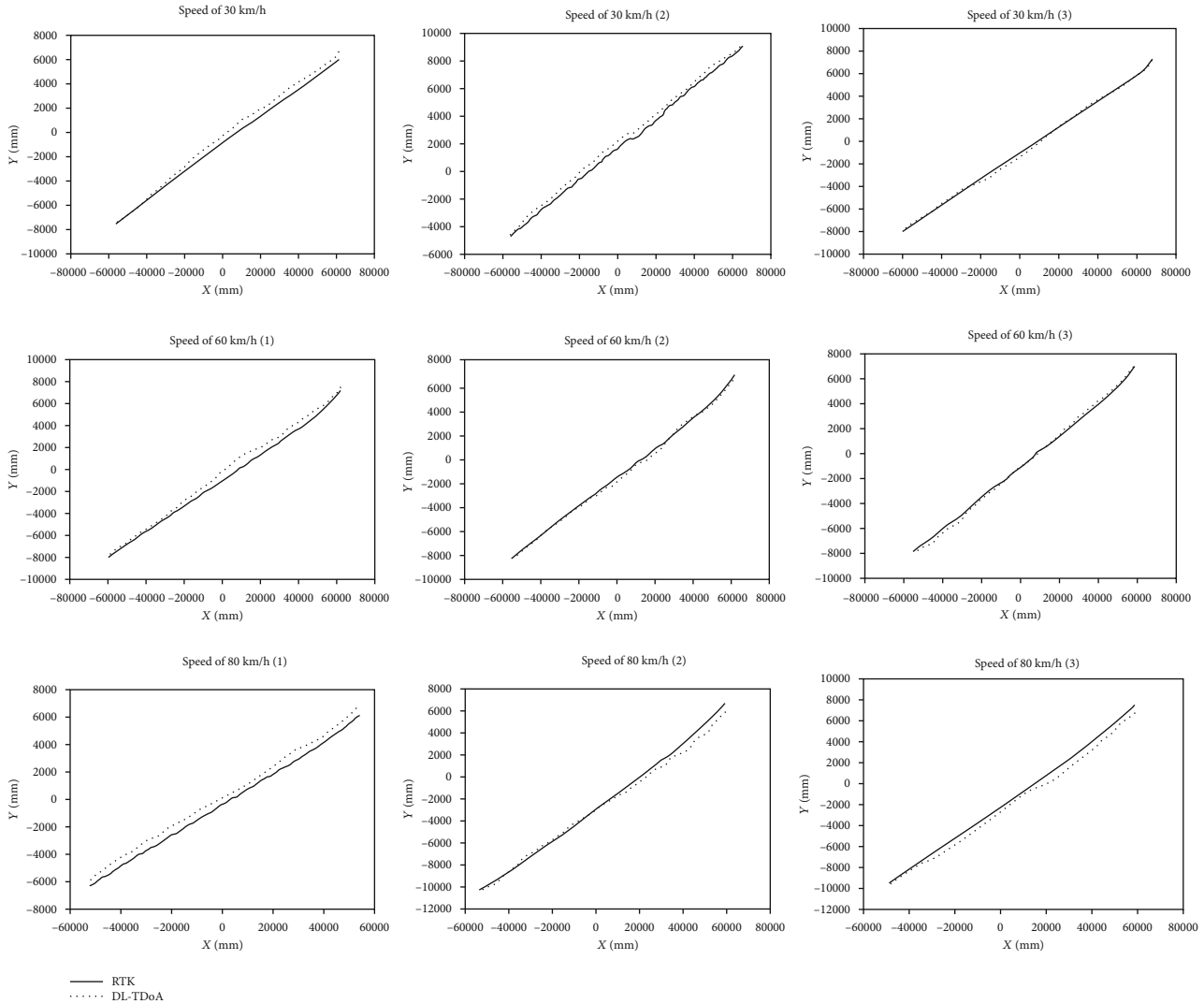


Figure 10: The comparisons of RTK and DL-TDoA with the speed of 30, 60, and 80 km/h, respectively.

RTK and DL-TDoA is different so testers need to manually find the same frame that both RTK and DL-TDoA have uploaded the data for the future comparisons and analysis. Figure 10 shows the comparisons of 30, 60, and 80 km/h, respectively, and the initial positions are found by testers so there might be some incorrect initial points. However, what we need to focus on is the trend of the lines, whether it is consistent with the RTK of the comparison sample. For the first and second figures of Figure 10 of 30 km/h, DL-TDoA has the same trend with RTK so the tester might not find the correct point but the results are still good. As for the third figure of Figure 10 (1), of 30 km/h, the DL-TDoA data nearly overlaps the RTK data but with a little fluctuation. That means testers find the correct initial point, and within the speed of 30 km/h, DL-TDoA performs as well as RTK. In the figures of 60 km/h in Figure 10, (1) shows that RTK has more lag fluctuation than DL-TDoA while the other two ((2) and (3)) show the DL-TDoA performs well. As for (1) of 80 km/h of Figure 10, it is obvious that the testers find a wrong initial position of DL-TDoA. However, compared with 30 and 60 km/h, the DL-TDoA positioning algorithm under the speed of 80 km/h performs worst. In conclusion, the higher the speed of the vehicles, the lower the accuracy of the test.

Table 2: Maximum, minimum, and mean of errors of 30, 60, and 80 km/h, respectively.

Speed (km/h)	Maximum error (mm)	Minimum error (mm)	Mean error (mm)
30	742.81	101.25	355.03
	775.68	117.57	418.85
	623.41	7.13	246.60
60	1126.77	45.41	961.23
	647.23	20.96	302.65
	845.97	52.01	205.75
80	1958.08	405.62	1306.81
	1496.90	53.26	951.28
	1493.73	62.79	789.23

Table 2 proves that when the speed of vehicles goes up, the maximum errors and mean errors increase. However,

Table 3: The average mean of errors of 30, 60, and 80 km/h, respectively, and delay.

Test scenario	Average mean error comparing with RTK (cm)	Positioning frequency (Hz)	Delay (ms)
Static	11.96	10	<50
30 km/h	37.11	26	<50
60 km/h	50.06	26	<50
80 km/h	87.03	26	<50

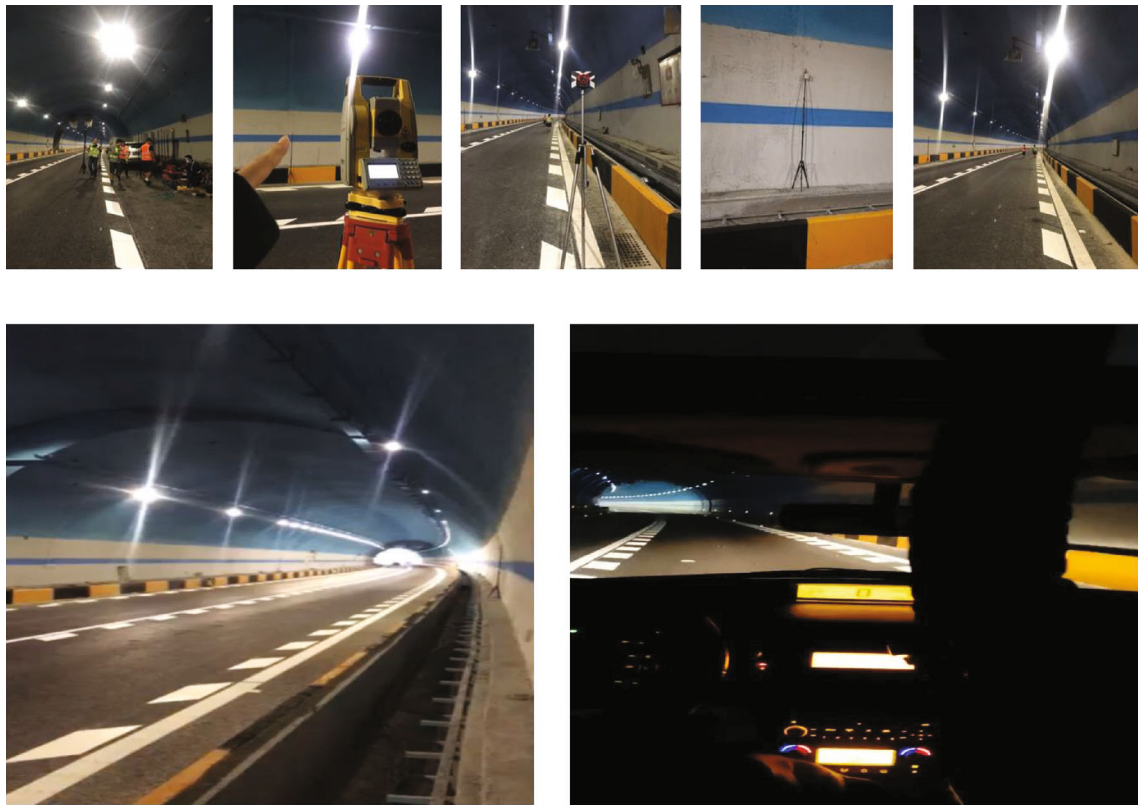


Figure 11: The tunnel scenario was set up.

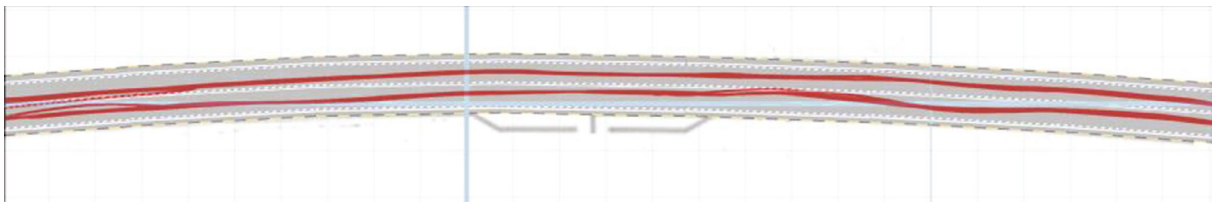


Figure 12: Tunnel test result by using DL-TDoA.

the minimum errors do not depend on speed. To ensure the safety and accurate positioning, the maximum errors are the most important data this paper needs to focus on. In conclusion in Table 3, when the vehicle goes through the test area with 30, 60, and 80 km/h, the mean errors are 37.11, 50.06, and 87.03 cm, respectively, which give an acceptable and useful result for future tunnel positioning without RTK comparison.

4.3. Scenario 2: Tunnel Test. In Figure 11, testers set up the scenario in a tunnel and use the same devices and sensors in the dynamic test. One driver drives the car and goes through the tunnel. Due to the unavailable RTK service, the data only comes from the DL-TDoA system. After converting the data to lines drawn on Figure 12 and checking with the real driving route, the detected trace is nearly the same accuracy as the highway dynamic test.

5. Conclusions

This paper proposed a DL-TDoA technology inherited from UWB wireless protocol to realize accurate vehicle positioning systems in highway and tunnel scenarios. Compared with

traditional positioning technology, the most important advantages of DL-TDoA technology are its high time efficiency and strong anti-interference capability during data transmission. Without a cumbersome connection process, DL-TDoA utilizes simple devices and distributed processing methods to realize a reliable, large capacity, and scalable wireless network. As shown by the test results in two scenarios, regarding RTK as the standard reference position, the location accuracy from DL-TDoA is less than 1 meter and frequency keeps stable at 4 Hz, 26 Hz. Also, the coverage rate of the DL-TDoA system achieves 100% in the tunnel scenario, which means all areas of the tunnel will be covered by the signals. In the tunnel scenario, the positioning accuracy of centimeter level can be obtained and the dynamic return can be achieved even when the vehicle system is driving at high speeds. In the future, other kinds of complex road scenes will be tested by the proposed DL-TDoA system.

Data Availability

The data can be available upon request to the corresponding author.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program (2019-0-01585 and 2020-0-01576) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). This research was funded by Beijing Capital Road Development Group Co., LTD., Beijing, China, and Beijing Sutong Technology Co., Ltd., Beijing, China, grant numbers "52200401-2-02 and 52200401-2-03." The APC was funded by Roadway Smart (Beijing) Technology Co., Ltd.

References

[1] S. Ghosh, S. Rao, and B. Venkiteswaran, "Sensor network design for smart highways," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 5, pp. 1291–1300, 2012.

[2] X. Xu, J. Yu, Y. Zhu, Z. Wu, J. Li, and M. Li, "Leveraging smartphones for vehicle lane-level localization on highways," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1894–1907, 2018.

[3] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 195–207, 1993.

[4] 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.

[5] 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.

[6] C. Liu, K. T. Chau, D. Wu, and S. Gao, "Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies," *Proceedings of the IEEE*, vol. 101, no. 11, pp. 2409–2427, 2013.

[7] S. Zhou and J. K. Pollard, "Position measurement using Bluetooth," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, pp. 555–558, 2006.

[8] F. Forno, G. Malnati, and G. Portelli, "Design and implementation of a Bluetooth ad hoc network for indoor positioning," *IEE Proceedings - Software*, vol. 152, no. 5, pp. 223–228, 2005.

[9] Y. Tao and L. Zhao, "A novel system for WiFi radio map automatic adaptation and indoor positioning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10683–10692, 2018.

[10] C. Y. Yao and W. C. Hsia, "An indoor positioning system based on the dual-channel passive RFID technology," *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4654–4663, 2018.

[11] S. H. Fang, C. H. Wang, T. Y. Huang, C. H. Yang, and Y. S. Chen, "An enhanced ZigBee indoor positioning system with an ensemble approach," *IEEE Communications Letters*, vol. 16, no. 4, pp. 564–567, 2012.

[12] Z. Yin, X. Jiang, Z. Yang, N. Zhao, and Y. Chen, "WUB-IP: a high-precision UWB positioning scheme for indoor multiuser applications," *IEEE Systems Journal*, vol. 13, no. 1, pp. 279–288, 2019.

[13] X. Shan and Z. Shen, "Miniaturized UHF/UWB tag antenna for indoor positioning systems," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 12, pp. 2453–2457, 2019.

[14] M. Eric and D. Vucic, "Method for direct position estimation in UWB systems," *Electronics Letters*, vol. 44, no. 11, pp. 701–703, 2008.

[15] M. Hamalainen, J. Iinatti, V. Hovinen, and M. Latva-aho, "In-band interference of three kinds of UWB signals in GPS L1 band and GSM900 uplink band," in *12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC 2001. Proceedings (Cat. No.01TH8598)*, San Diego, CA, USA, 2001.

[16] A. Zhou, S. Wang, S. Wan, and L. Qi, "LMM: latency-aware micro-service mashup in mobile edge computing environment," *Neural Computing and Applications*, vol. 32, no. 19, pp. 15411–15425, 2020.

[17] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[18] E. Wang, Z. Zhang, and M. Cai, "A study on BeiDou positioning performance analysis platform based on LabVIEW," in *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 963–967, Melbourne, VIC, Australia, 2013.

[19] D. V. Nguyen, F. Nashashibi, T. K. Dao, and E. Castelli, "Improving poor GPS area localization for intelligent vehicles," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 417–421, Daegu, South Korea, November 2017.

[20] A. Bhaskar, M. Qu, and E. Chung, "Bluetooth vehicle trajectory by fusing Bluetooth and loops: motorway travel time statistics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 113–122, 2015.

[21] W. Albazrqaoe, J. Huang, and G. Xing, "A practical Bluetooth traffic sniffing system: design, implementation, and

countermeasure," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 71–84, 2019.

[22] B. Song, J. Y. Hwang, and K. A. Shim, "Security improvement of an RFID security protocol of ISO/IEC WD 29167-6," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1375–1377, 2011.

[23] Y. Y. Shih, W. H. Chung, P. C. Hsiu, and A. C. Pang, "A mobility-aware node deployment and tree construction framework for ZigBee wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2763–2779, 2013.

[24] J. F. Gerrits, J. R. Farserotu, and J. R. Long, "Low-complexity ultra-wide-band communications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 4, pp. 329–333, 2008.

[25] P. du, S. Zhang, C. Chen, A. Alphones, and W. D. Zhong, "Demonstration of a low-complexity indoor visible light positioning system using an enhanced TDOA scheme," *IEEE Photonics Journal*, vol. 10, no. 4, pp. 1–10, 2018.

[26] J. Wang, X. Qiu, and Y. Tu, "An improved MDS-MAP localization algorithm based on weighted clustering and heuristic merging for anisotropic wireless networks with energy holes," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 227–244, 2019.

[27] I. Um, S. Park, H. T. Kim, and H. Kim, "Configuring RTK-GPS architecture for system redundancy in multi-drone operations," *IEEE Access*, vol. 8, pp. 76228–76242, 2020.

[28] K. M. Ng J. Johari et al., "Performance evaluation of the RTK-GNSS navigating under different landscape," in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1424–1428, Daegwallyeong, 2018.

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

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

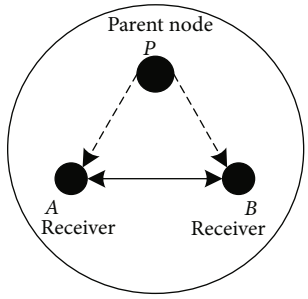


Figure 1: Schematic diagram of the RBS synchronization mechanism.

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 $200 \le n \le 500$) 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

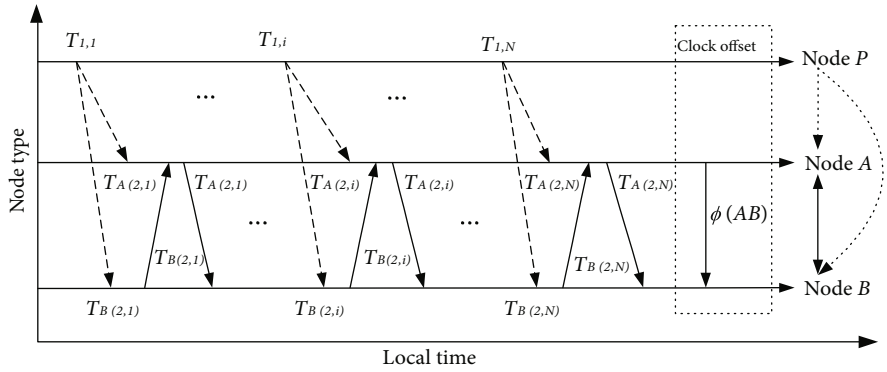


Figure 2: Schematic diagram of clock synchronization RBS model.

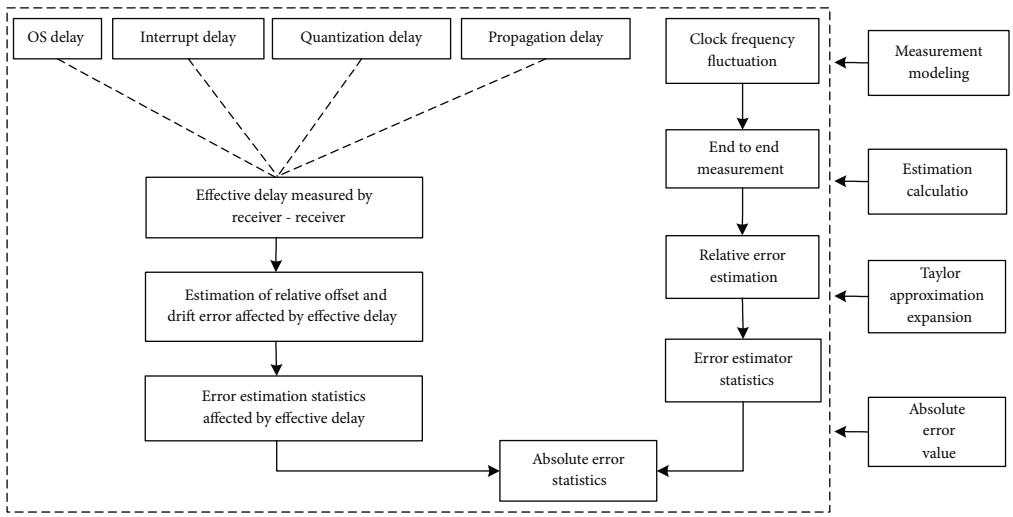


Figure 3: Analysis of synchronization errors.

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].

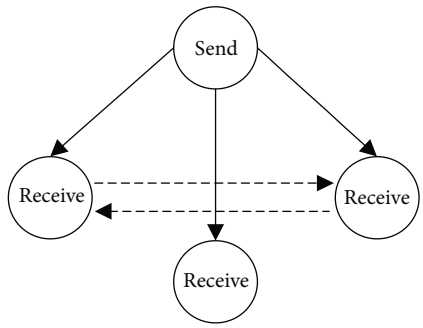


Figure 4: Diagram of the estimation of the phase offset of the ERBS algorithm.

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

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

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

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

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 $f_i(t) = \nu$, we can get a more simplified time model.

$$y = \nu x + \sigma, \tag{16}$$

where ν 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.

$$\nu = \bar{y} - \sigma\bar{x}, \tag{17}$$

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

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

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

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

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

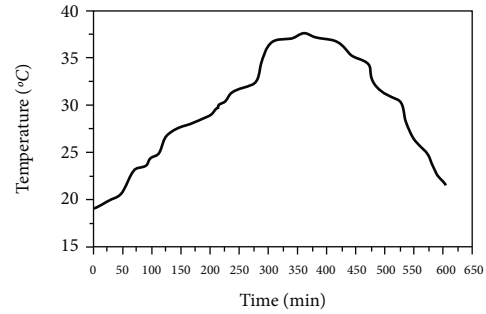


Figure 5: Curve showing changes in temperature over time.

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:

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

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

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:

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

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

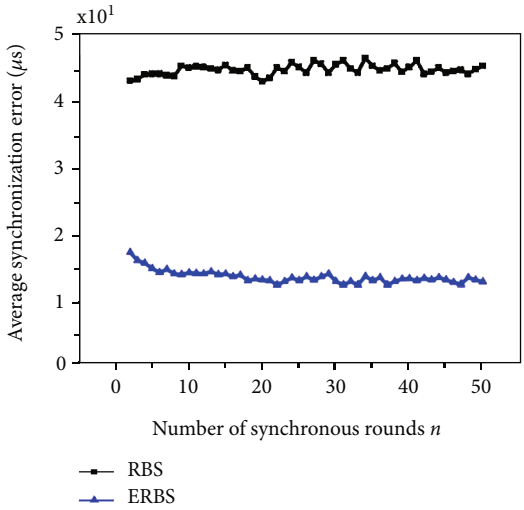


Figure 6: Curves of average synchronization errors (opening timestamps on the MAC layer).

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

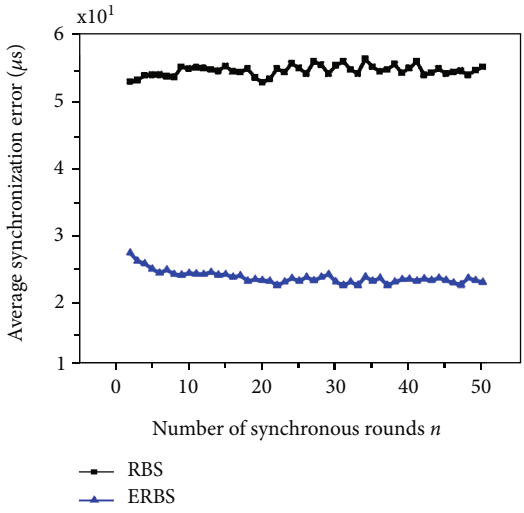


Figure 7: Curves of average synchronization errors (closing the timestamps at the MAC layer).

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

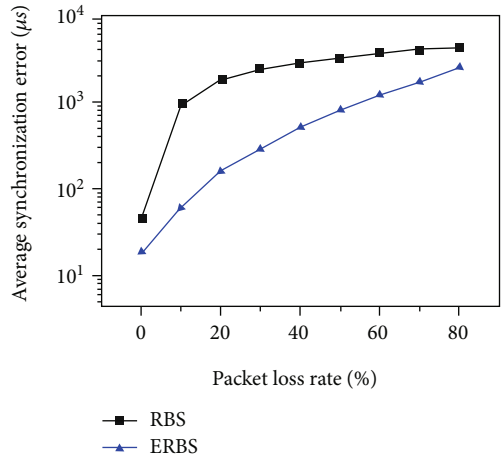


Figure 8: Time synchronization errors in cases of different packet loss rate.

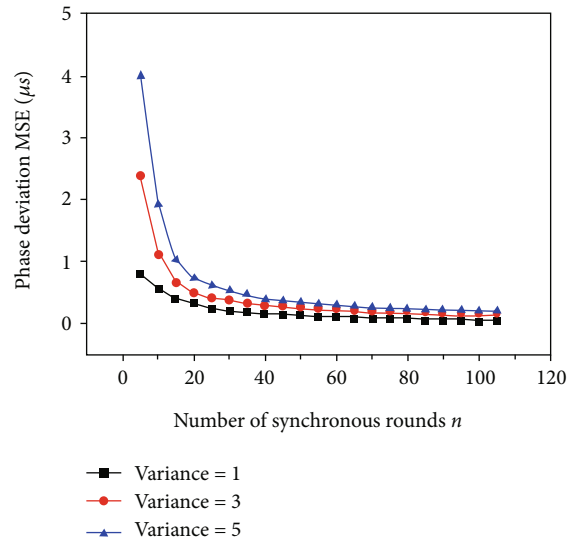


Figure 9: MSE curves of clock phase offset.

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

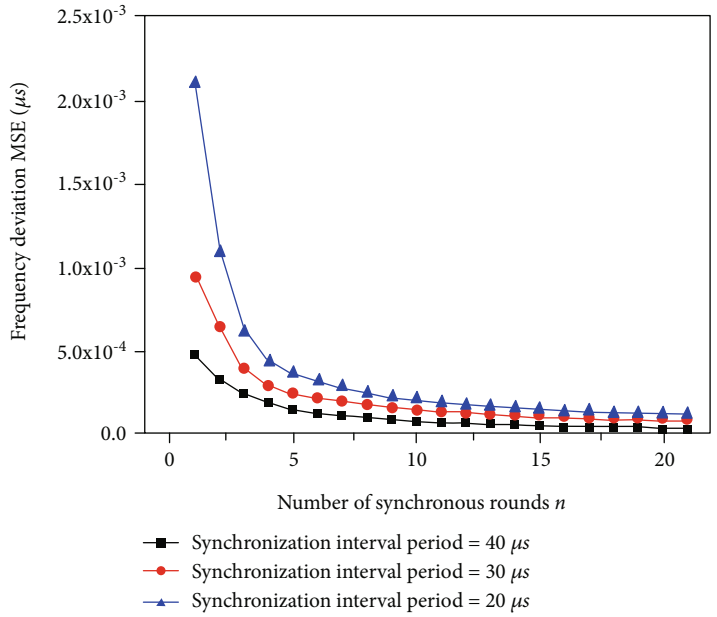


Figure 10: MSE curves of clock frequency offset (round of synchronization n).

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.

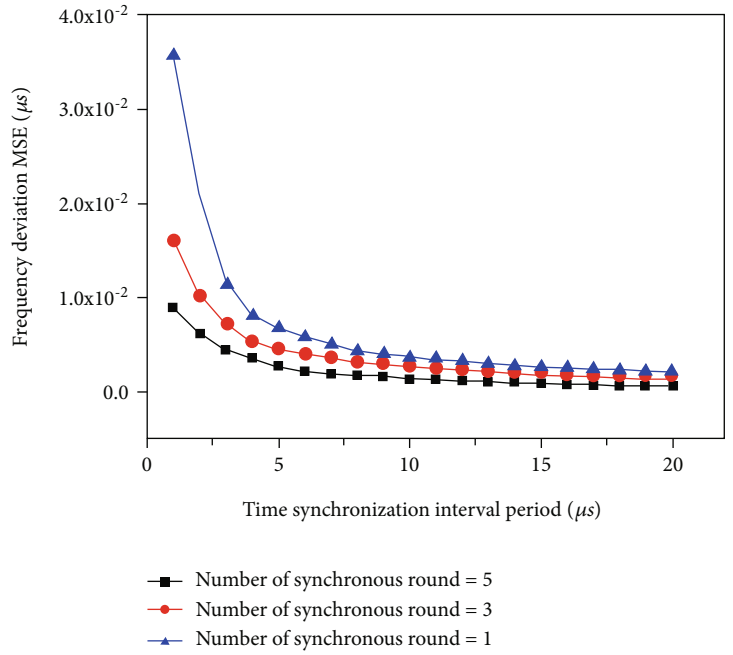


Figure 11: MSE curves of clock frequency offset (synchronous interval cycle τ).

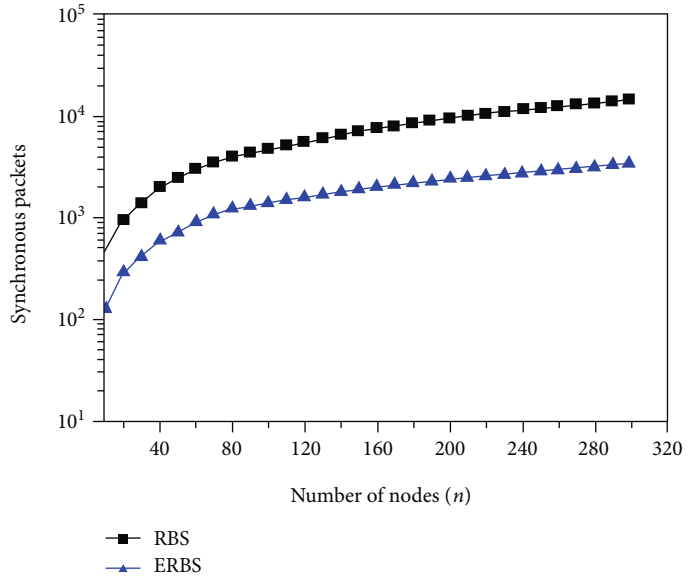


Figure 12: The relationship curve between synchronization data packets and the number of nodes.

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

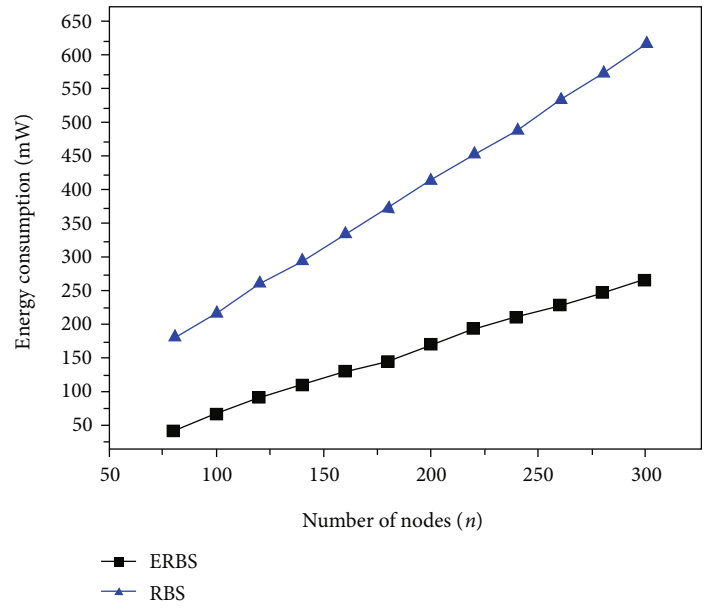


Figure 13: Comparison of energy consumption.

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.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8869898, 13 pages
https://doi.org/10.1155/2020/8869898

Research Article

A Packet Scheduling Method Based on Dynamic Adjustment of Service Priority for Electric Power Wireless Communication Network

Bo Hu,[1] Xin Liu,[2] Jinghong Zhao,[3] Siya Xu,[2] Zhenjiang Lei,[1] Kun Xiao,[2] Dong Liu,[3] and Zhao Li[1]

[1]*State Grid Liaoning Electric Power Supply Co., Ltd., Shenyang 110000, China*
[2]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*Information and Communication Branch, State Grid Liaoning Electric Power Co., Ltd., Shenyang Liaoning 110000, China*

Correspondence should be addressed to Xin Liu; lxliuxin@bupt.edu.cn

Received 1 September 2020; Revised 7 October 2020; Accepted 20 October 2020; Published 2 November 2020

Academic Editor: Shaohua Wan

With the development of the energy Internet, power communication services are heterogeneous, and different power communication services have different business priorities. The power communication services with different priorities have different requirements for network bandwidth and real-time performance. For traditional unified service, a scheduling method cannot meet these service requirements at the same time, and electric power communication network cannot guarantee the quality of service. Therefore, how to make full use of the time-varying characteristics of communication resources to meet the business needs of different priorities and achieve the goals of high resource utilization and transmission quality has become one of the urgent problems in the power communication network. For this reason, in order to adapt to the real-time congestion of the network, we have designed a packet scheduling method based on the dynamic adjustment of service priority, which dynamically adjusts the priority of the power service on the node; in addition, an evaluation method for the trust value of wireless forwarding nodes is introduced to improve the security of data transmission; and finally, we valuate the channel quality to establish a reasonable and efficient packet scheduling mechanism for services of different priorities. Simulation results show that this method improves the communication performance of high-priority services and improves the spectrum resource utilization of the entire system.

1. Introduction

With the development of energy Internet, based on the traditional power communication facilities, a large number of mobile heterogeneous terminal nodes are distributed on the edge of the network, forming a mobile self-organizing network. Mobile ad hoc network is a kind of distributed wireless ad hoc network. Data packets are transmitted in the way of multihop forwarding in the connected links of the network. With flexible architecture, it is used to carry power communication network services, which has become a trend in the future.

Due to the continuous development of intelligent power distribution network and the increase of distribution and consumption communication service types and traffic, the existing wireless network resources cannot adapt to the large-scale deployment of smart grid. The QoS indicators of power communication services include data rate, delay, and packet loss rate. By ensuring transmission bandwidth and reducing transmission delay, packet loss rate, and delay jitter, the quality of service can be improved. Compared with the general communication network, the service of power communication network is more heterogeneous, and the quality of service (QoS) requirements of different services are also

very different. Control service has high requirements for delay and reliability [1–4]. For example, distribution automation and distributed power control services need to have channel protection and communication load control functions, so it is necessary to establish low delay and high data rate communication between substation and intelligent power equipment; power information collection service of users does not require high real-time performance and transmission rate [5, 6], but the communication volume is large, and the information security requirements are high.

Smart grid introduces the concept of dynamic adjustment of service priority to solve the problems of spectrum resource shortage and low spectrum utilization rate faced by smart grid [7–9]. For power communication service priority adjustment, the existing methods cannot dynamically adjust the priority level of data packets according to the real-time congestion state of the network; for the scheduling of service packets, the traditional scheduling mechanism lacks consideration of channel resource changes and cannot provide reliable quality assurance for low-priority services under the condition of real-time changes of available channel transmission resources [10, 11]. Therefore, the improved routing algorithms include on-demand QoS routing algorithm, QoS routing algorithm based on fuzzy control, and opportunistic scheduling algorithm with packet delay guarantee [12]. In addition, most of the existing packet scheduling mechanisms only consider the absolute priority of high-priority services, ignoring the relative priority of other services, which cannot meet the requirements of providing differentiated QoS services for heterogeneous services in the intelligent power communication network.

Therefore, how to use efficient service scheduling algorithm to make full use of the time-varying characteristics of communication resources, to support the future intelligent distribution and utilization of power communication edge data transmission, and to meet the requirements of high resource utilization and transmission quality, has become one of the problems to be solved urgently in power communication network. Therefore, this paper studies a dynamic priority-based power wireless communication service packet scheduling mechanism to adapt to the operation requirements of the future intelligent power distribution and utilization communication network.

The main contributions of this article are as follows:

(1) According to the communication requirements and service characteristics of the intelligent power distribution and utilization wireless communication network, in consideration of the service QoS requirements and importance, we design a complete dynamic adjustment algorithm of power wireless communication service packet priority to adapt to the different states in the network operation and ensure the quality of service

Based on the dynamic adjustment algorithm of power wireless communication service packet priority, we comprehensively consider the real-time change of network transmission resources; then, an evaluation method for the trust value of wireless forwarding nodes is introduced to improve the security of data transmission; and finally, we valuate the channel quality to establish a reasonable and efficient packet scheduling mechanism for services of different priorities, ensure the QoS requirements of various services, and optimize the system utilization rate.

(2) The reminder of this article is organized as follows. In Section 4, the system model is proposed, and the packet scheduling mechanism of power communication service based on dynamic priority is discussed in Section 5. Simulation process is given and discussed in Section 6. Finally, we conclude this paper in Section 7

2. Related Work

At present, most researches on service routing are in-depth research on the above-mentioned key technologies including node mobility model, message forwarding mechanism, and congestion control mechanism. This section will introduce the following two mainstream routing algorithms and the latest research on mainstream algorithms.

2.1. Routing Based on Delivery Probability. Lindgren et al. proposed a routing strategy based on delivery probability [13, 14], namely, PRoPHET routing. The node records its own historical information, and when the nodes meet, they share each other's historical information and transfer information. Use this information to evaluate network information, thereby predicting the contact probability of a node to other nodes. Suppose that node S holds a message and its destination node is D. When node S encounters node B, if the probability of contact between node B and node D is greater, the router thinks that node B is more likely to deliver message m to node C. Node A copies and forwards the message to node B; otherwise, node S does not forward it. That is, in this routing mechanism, the transmission of service is more inclined and will be delivered to nodes that have greater contact with the destination node. Among them, the calculation of contact probability mainly has two parameters: one is the attenuation weight γ, and the other is the transfer weight β. The contact probability of two frequently connected nodes is updated and increased each time they meet. When two nodes do not meet within a certain time interval, the contact probability between the two nodes becomes smaller under the effect of the attenuation weight. If node S is in contact with node B, node B is in contact with node D, but node S is not in contact with node D, and the existence of the transfer weight will bring the probability of contact between S and D to a certain value. Although the PRoPHET method selects the node of the infected message, its copy amount in the network is still unlimited, and it is still easy to cause network congestion and cause network performance deterioration. For the vehicle opportunity network [15, 16], Du et al. combined the message transfer strategy of the PRoPHET protocol with the message replication control strategy of the jet waiting protocol, which effectively controlled the number of replications, thereby reducing the overhead. Bai et al. proposed a

Bayesian network-based method to estimate the contact probability between network nodes, which improves the accuracy of the contact probability estimation, thereby improving the performance of routing.

2.2. Routing Based on Different Business Requirements. The definition of business distinction can be divided into two types, one is the distinction of the same type of business due to different contents, such as the priority of rescue information when a disaster occurs. The other is business differentiation caused by different business types, such as text, picture, or video services.

Mashhad and Capra considered a general model for measuring the priority of messages [17] and modeled user's interest in messages in two ways: users can define the objects they are interested in (people-centered) or the content of interest (in content as the center). Regarding service differentiation caused by different service types, different types of services have different QoS requirements such as bandwidth, packet loss rate, and delay. The research focuses on the QoS guarantee of different services. Although the network provides three different QoS classes: accelerated, normal, and batch, to distinguish messages, if you simply prioritize messages based on their QoS class, applications that belong to the lower class will not be able to get it. To the transmission opportunity. Some research focuses on how to solve the sorting problem between different classes. Tajima et al. defined the data arrival rate as the data arrival rate of all data that reached the target node [18] and the rate of all data that was deleted due to timeout but did not reach the target node. It is called the data loss rate. Determine the data discarded by the node when the data in the buffer is full, estimate the data loss rate and data arrival rate of the entire network, and modify the buffer partition ratio according to the deletion rate of the priority category [19–21]. Xu et al. proposed the concept of reference probability. The meeting node defines different reference probabilities for different data packet priorities. If the reference probability of the meeting node is greater than the forwarding probability of the sending node, the data packet is forwarded; otherwise, it is not forwarded [22–24]. But these strategies do not adapt to the dynamic changes of the network. If resources are insufficient to meet all constraints, the strategy will still allocate resources proportionally and may not meet any category of requirements or even the highest priority requirements. On the other hand, if resources are sufficient, it may unnecessarily continue to favor higher-level classes, restricting lower-level classes to achieve high performance. Considering the above problems, Matzakos et al. proposed a routing algorithm to adapt to resource allocation in a dynamic environment [25–27] and defined constraints to optimize network-wide performance while satisfying the QoS constraints of a single category, but the contagion strategy it uses consumes a large resource consumption, and scheduling information needs to be provided globally [28, 29], which is not suitable for the actual application of delay-tolerant networks.

Therefore, the subject researches an opportunistic routing planning algorithm based on business priority in a dynamic network that can use local information.

The above research focuses on the services differentiated by content, and the research focuses on how to differentiate the services and determine their priorities. However, low-priority services may not get transmission opportunities for a long time, which is not always feasible in the actual power communication network. In order to fill this research gap, we focus on the priority dynamic adjustment algorithm for power communication services and propose a dynamic scheduling mechanism for service data packets to solve complex service routing problems.

3. System Model

In the intelligent distribution communication network involved in this chapter, the elements that need to be investigated include the services with different priorities in the network, network spectrum resources, and the network behavior (channel access, backoff, and handover).

3.1. Service Description Model. In the network, each service will have a service description model to describe its constraint information and application attribute information, as shown in Figure 1.

(1) Type: basic description of service, such as voice, text, picture, and important notice

(2) Size: for the description of service size, too large service description will cause the increase of network transmission cost, such as occupying too much cache space and occupying longer transmission time

(3) DelayGoal: if the current network can support the goal of delay, the algorithm will determine the minimum number of copies based on the target delay

(4) LossGoal: if the desired service quality of the service is to be achieved, the goal of packet loss rate of the service needs to be greater than the current network packet loss rate

(5) DalayAccept: if the current network cannot support the target delay of the service, the number of copies will be determined based on the maximum acceptance delay

(6) LossAccept: the target packet loss rate and maximum accepted packet loss rate, the target delay, and the maximum accepted delay form the service quality range of the service. If the network cannot support the demand for service quality, the service request can be rejected

(7) PriLevel: this indicates that the user wants the service level provided by the network. There are three priority levels: H, M, and L

(8) DownFlag: accepting the downgrade sign indicates that a certain quality of service can be downgraded when the network is congested

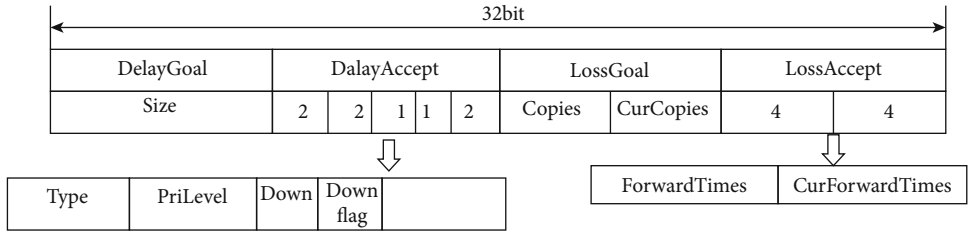


Figure 1: Service description model.

(9) Down: when the network is in congestion, it is set to 1 to indicate that the service is downgraded. When the network is in a good state, down is set to 0

(10) Copies: the maximum number of copies of service description. The more the copies of service description, the larger the buffer space occupied, and the more the network resources required. So we need to preallocate the number of copies of service description

(11) CurCopys: it represents the number of copies of the service description owned by the current node

(12) CurForwardTimes: the number of times the service description was forwarded during the wait step

(13) ForwardTimes: the number of times the service description can be forwarded during the wait step

3.2. The Specific Priority of Service. Electric power communication network services can be divided into production control area services and management information area services according to their categories. The specific priority of service is shown in Table 1.

3.3. Network Spectrum Resources. The resource of cognitive wireless network is network spectrum resource. Since each service is assigned an authorized channel, the channel set can be represented as $\mathbf{F} = \{F_1, F_2, \cdots F_M\}$.

3.4. Network Behavior. Each service uses idle spectrum resources for data transmission. However, if high-priority service reappears during low-priority service information transmission, low-priority service should be immediately discarded from the channel or switched to other channels to continue transmission. Therefore, low-priority service's network connectivity is affected by high-priority service's behavior, and its services are often in the state of interruption or switching. The spectrum of low-priority service available links is different and dynamic due to the activity of high-priority service.

4. Algorithm Design

4.1. Service Priority Dynamic Adjustment Strategy. In order to maximize the delivery rate, this section mainly solves two problems related to spray wait routing algorithm.

(1) When multiple messages coexist in the local buffer and the node is unable to determine whether the node and the meeting node will forward all messages long enough, it should decide which message to send first

(2) If a new message arrives at the node's buffer and overflows, a drop decision should be made between messages, that is, which message to discard

In order to solve the above two problems, we set a priority for messages in the internal queue of the node to determine the ordering and discarding of messages. The queue priority of messages in a node is a complex function of the number of message copies, the remaining number, and the priority of the message itself [17].

$$\text{Priority}_i = \Delta P = f(\text{TTL}_i, C_i, W_i). \tag{1}$$

Table 2 shows the parameters and explanations involved in this section.

The meeting time of a node with other $N-1$ nodes in the network is I_i, $i \in \{1, 2, 3, \cdots, N\text{-}1\}$. The encounter time satisfies the exponential distribution, and the parameter is λ_e [30]. Therefore, the minimum encounter time is $\boldsymbol{I}_{\min} = \min_{\boldsymbol{i} \in \{1,2,3,\cdots,\boldsymbol{N}-1\}}$,

$$\lambda_{\min} = \frac{1}{E(I_{\min})} = \frac{N\text{-}1}{E(I)}. \tag{2}$$

The calculation [30] of $P(i)$ is shown in formula (3). The delivery probability of message i is composed of the probability $P(T_i)$ that message i has been delivered and the probability $P(RT_i)$ that message i will be delivered within the remaining time RT_i.

$$\boldsymbol{P}(\boldsymbol{i}) = \boldsymbol{P}(\boldsymbol{T_i}) + (\boldsymbol{1} - \boldsymbol{P}(\boldsymbol{T_i}))\boldsymbol{P}(RT_{\boldsymbol{i}}). \tag{3}$$

Assume that the number of nodes that receive message i be $m_i(T_i)$ and the number of nodes currently holding the message i be $n_i(T_i)$; the probability$P(T_i)$that message$i$has been delivered is:

$$\boldsymbol{P}(\boldsymbol{T_i}) = \frac{\boldsymbol{m_i}(\boldsymbol{T_i})}{\boldsymbol{N}-1}, \tag{4}$$

where 1-$P(RT_i)$ means that the message i is delivered not only in T_i but also in the remaining time $RT(\text{TTL}_i - T_i)$

Table 1: Electric power wireless communication business priority.

Power communication network services division		Specific services	Services priority
Production control area services	Control area services	Energy management system (EMS), relay protection system, security automatic control system, etc.	First-level services
		Emergency power demand response system	Second-level services
	Noncontrol area services	Electric energy metering system, relay protection and fault recording information management system, etc.	Third-level services
Management information area services		Management information system (MIS), office automation system (OA), customer service system, etc.	Fourth-level services

Table 2: Network parameters.

Symbol	Description
$DN(t)$	Total number of messages in the network (excluding copies)
C	Maximum number of copies of a message
C_i	Number of copies of message i on the node
U_i	Priority of message i on the node
W_i	Service priority weight for message i
W_d	Service priority weight
I	Time when a node meets another node in the network, following an exponential distribution $f(x) = \lambda_e e^{-\lambda_e x}$
$E(I)$	Expectation of meeting time
λ_e	Exponential distribution parameter for meeting time ($\lambda_e = 1/E(I)$)

probability that will pass. Assume that RT_i is long enough to spray all copies. The C_i copy of message i will continue to be transmitted to the $\log_2 \boldsymbol{C_i}$ node until the number of copies is reduced to 1. In addition, the interval between adjacent infections can be estimated as $E(I_{\min})$. Every $E(I_{\min})$ time unit, one node will receive the message. W_i is the initial business priority weight of the message i, and W_d is the downgrade weight. (RT_i) can be expressed as follows:

$$P(RT_i) = 1 - \prod_{k=0}^{\log_2 C_i} e^{-\lambda_e n_i(T_i)[W_i W_d RT_i - kE(I_{\min})]} = 1 - e^{-\lambda_e n_i(T_i)[(\log_2 C_i + 1) W_i W_d RT_i - (1/(2(N-1)\lambda_e))\ \log_2 C_i (\log_2 C_i + 1)]}. \tag{5}$$

Combine the above formulas:

$$P(i) = \frac{m_i(T_i)}{N-1} + \left(1 - \frac{m_i(T_i)}{N-1}\right) \cdot \left(1 - e^{-\lambda_e n_i(T_i)[\log_2 C_i + 1] W_i W_d RT_i - (1/(2(N-1)\lambda_e))\ \log_2 C_i (\log_2 C_i + 1)}\right). \tag{6}$$

Global success rate P is calculated as

$$P = \sum_{i=1}^{k(t)} P(i). \tag{7}$$

Derivative is calculated as

$$\Delta P = \sum_{i=1}^{k(t)} \left[\frac{\vartheta P}{\vartheta n_i(T_i)} \Delta n_i(T_i)\right]. \tag{8}$$

The priority adjustment queue proposed in this section is to maximize the delivery rate. When nodes meet, cancel the service i; if the injection is carried out, the number of nodes holding the message in the network will increase, $\Delta n_i(T_i) = 1$; if not, the number of nodes holding the message will increase. If there is injection, the number of nodes holding the message does not change, $\Delta n_i(T_i) = 1$. The priority adjustment queue proposed in this section is to maximize the delivery rate. The priority of message i is exactly the derivative of the delivery rate P.

$$U_i = \left(1 - \frac{m_i(T_i)}{N-1}\right)\lambda_e \cdot \left[(\log_2 C_i + 1) W_i W_d RT_i - \frac{1}{2(N-1)\lambda_e}\ \log_2 C_i (\log_2 C_i + 1)\right] \times e^{-\lambda_e n_i(T_i)[(\log_2 C_i + 1) W_i W_d RT_i - (1/(2(N-1)\lambda))\ \log_2 C_i (\log_2 C_i + 1)]}. \tag{9}$$

The calculated priority is a composite function of the number of message copies, the remaining TTL, and the service priority of the message itself, which can estimate the message more accurately. In most cases, the large number

of remaining copies of messages and the remaining TTL indicate that the scope of message infection is small, and these messages should have higher priority.

Each node can calculate the priority of messages in the buffer. Therefore, the node can schedule the sending order and make the discard decision according to the priority. Each node manages its buffer in a distributed way, which means that each node only cares about the priority in its own buffer. When two nodes meet, they only consider which message to send between messages in the buffer and which message to delete when the overflow occurs.

In formula (10), $m_i(T_i)$ is the number of nodes that receive message i; $n_i(T_i)$ is the number of nodes currently holding the message now. Assume that $d_i(T_i)$ is the number of nodes that have discarded the message:

$$n_i(T_i) = m_i(T_i) + 1\text{-}d_i(T_i). \tag{10}$$

In order to accurately estimate $d_i(T_i)$, each node maintains one piece of information of the discard history, including node ID, list of discarded messages, and record collection time.

Assume that the size of the above data structure is negligible compared to the size of the message. The discard queue contains all discarded message ID, and the record time is the generation time of the record. When nodes meet, they exchange and update their records. Only the source node can modify the record time and only if a new discard occurs in its buffer. When two nodes meet, exchange and update their respective discard history information. After a period of time, each node can estimate $d_i(T_i)$.

The estimation of $m_i(T_i)$ is obtained by the binary characteristic of the binary spray and waiting routing algorithm. The binary spray and waiting routing algorithm is shown in Figure 2. In the whole process, the time when the message is sprayed is recorded, so we can estimate the message transmission process of each node.

The current number of copies of message i is C_i, the initial number of copies is c, and then the height of the tree can be obtained.

$$h = \log_2 \frac{c}{C_i}. \tag{11}$$

Messages are sprayed into a binary tree after a period of time.

$$m_i(T_i) = \sum_{k=1}^{h-1} 2^{[(t_n - t_k)/E(I_{\min})]} + 1. \tag{12}$$

4.2. Trust Value of the Node. For secure communication, only nodes with high trust values should be selected for communication. If the computing task is forwarded by a wireless node with a low trust value, the node may take malicious actions, such as discarding data packets. Therefore, every mobile device should interact with a wireless with a high trust value to avoid potential security threats. In order to calculate the trust value of the forwarding node, we introduce the node trust value for evaluation. In this article, we use real numbers between 0 and 1 to evaluate the trust value of cooperative user nodes.

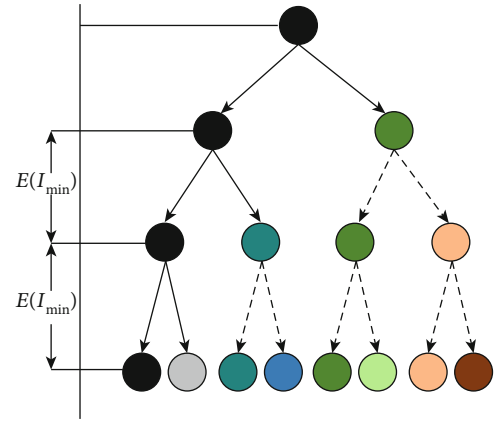


Figure 2: Spraying binary tree process.

Similar to [31], we use node honesty and node capacity to calculate direct trust. Since the mobile communication channel between the mobile device and the forwarding node is unstable and noisy, the communication behavior of the wireless node has considerable uncertainty. We use a subjective logic framework to deal with uncertainty. In the subjective logic framework, the trust value of the mobile device n to the mobile node x_{k_i} can be described as the triple $\omega_{n\to k} = \{b_{n\to k}, d_{n\to k}, v_{n\to k}\}$, where $b_{n\to k}$, $d_{n\to k}$, and $v_{n\to k}$ represent trust, distrust, and uncertainty, respectively. In particular, the relationship between them is determined by the following formula:

$$\begin{aligned} &b_{n\to k}, d_{n\to k}, v_{n\to k} \in [0, 1], \\ &b_{n\to k} + d_{n\to k} + v_{n\to k} = 1. \end{aligned} \tag{13}$$

Based on the trust model of [32], node honesty (NH) can be given by the following formula:

$$\mathrm{NH}_{n\to k} = b_{n\to k} + \xi v_{n\to k}, \tag{14}$$

where $0 \le \xi \le 1$ is a constant representing the degree of influence of trust uncertainty, and

$$\begin{aligned} b_{n\to k} &= (1 - v_{n\to k}) \frac{\alpha_{n\to k}}{\alpha_{n\to k} + \beta_{n\to k}}, \\ d_{n\to k} &= (1 - v_{n\to k}) \frac{\beta_{n\to k}}{\alpha_{n\to k} + \beta_{n\to k}}, \\ v_{n\to k} &= 1 - l_{n\to k}, \end{aligned} \tag{15}$$

where $\alpha_{n\to k}$ and $\beta_{n\to k}$ are the number of successful and failed communications, respectively. $l_{n\to k}$ represents the quality of the communication link, which refers to the probability of packet success. Packet loss is caused not only by the mobile communication channel but also by malicious nodes.

Therefore, the values of $\alpha_{n\to k}$ and $\beta_{n\to k}$ can be recalculated as

$$\begin{aligned}\alpha_{n\to k}^{\text{new}} &= \alpha_{n\to k} + P_{n\to k}^{\text{plr}} \times (\alpha_{n\to k} + \beta_{n\to k}),\\ \beta_{n\to k}^{\text{new}} &= \beta_{n\to k} - P_{n\to k}^{\text{plr}} \times (\alpha_{n\to k} + \beta_{n\to k}),\end{aligned} \tag{16}$$

where $P_{n\to k}^{\text{plr}}$ is the packet loss rate. Similar to [31], the packet loss rate is estimated by the following formula:

$$P_{n\to k}^{\text{plr}} = 1 - \frac{\sum_b^c \omega(b) \times \omega(b)}{\sum_b^c \omega(b)}, \tag{17}$$

where $\omega(b)$ is the weight value of the historical link state, and let $\text{link} = (\omega(1), \omega(2), \cdots, \omega(b))$ be the historical link state record. The weighted value is given by $\omega(b) = 2b/c(c+1)$, where b and c are the serial number and status record number of $\omega(b)$ in the link, respectively.

On the other hand, we assume that all wireless nodes have the same initial energy consumption rate and energy level. When a malicious node launches a malicious attack, it can always consume abnormal energy. Note that the initial energy consumption level of the nodes is the same. Therefore, we measure the trust of the node by the degree of change in the energy consumption level and judge whether the node is a malicious node. Let $P_{n\to k}^{\text{pen}}$ be the energy consumption rate, which is achieved by using the ray projection method [33] ($P_{n\to k}^{\text{pen}} \in [0, 1]$). Then, the node capability (NC) is given by

$$\text{NC}_{n\to k} = \begin{cases} 1 - P_{n\to k}^{\text{pen}}, & \text{if } E_{n\to k}^{\text{res}} \geq \theta,\\ 0, & \text{otherwise,}\end{cases} \tag{18}$$

where $E_{n\to k}^{\text{res}}$ and θ are the remaining energy and energy threshold of a node, respectively.

Direct trust values of nodes are calculated based on subjective logic; in this work, we evaluate the trust value of a node by a real number ranging from 0 to 1. Like most literature, such as [34, 35], the trust threshold is set 0.5. In other words, the node is trustworthy when its trust value is higher than 0.5; otherwise, it is not trustworthy. Then, the direct trust of the node is defined as

$$D_{n\to k}^{\text{direct}} = \begin{cases} 0.5 + (\text{NH}_{n\to k} - 0.5) \times \text{NC}_{n\to k}, & \text{if } \text{NH}_{n\to k} \geq 0.5,\\ \text{NH}_{n\to k} \times \text{NC}_{n\to k}, & \text{otherwise.}\end{cases} \tag{19}$$

To avoid potential security risks, we should only choose nodes with high trust values for communication. Each mobile device interacts with a node with a high degree of trust to obtain the trust value of each node and the priority of the service. The higher priority service should choose the node with the higher trust value to communicate for safer service communication.

4.3. Channel Quality Assessment. In power communication networks, we assume that different channels have the same bandwidth in the initial state, but processing services will occupy a certain amount of channel bandwidth resources. The routing algorithm metrics for communication services mainly include link stability, channel switching times, time delay, and bandwidth [12]. In addition to the above factors, the impact of high-priority services on low-priority services should also be examined, such as frequent link interruption and reestablishment and interference released by lower-priority services when higher-priority services occupy channels. In order to comprehensively evaluate the channel quality, three concepts of channel connectivity, reliability, and stability are defined as parameters to measure channel quality.

(1) Connectivity: system connectivity indicates whether the channel is connected between the user node and the base station at the current scheduling time. When the value of $L_{i,j}$ is 1, it means that the channel F_j is available for service; when it is 0, it means that the channel is not available

(2) Reliability: whether a channel is available for power communication services depends not only on connectivity but also on whether the channel is interfered by other services, sensor errors, and channel switching

Interference received from higher priority services: In the power communication network, if a higher priority service appears, the current service transmitted on the channel will be interfered by the higher priority service. Let $G_{i,j}$ be the probability that the current service is interfered by higher priority services in channel F_j; then, the probability of not being interfered can be expressed as

$$1 - G_{i,j} = \prod_{(x>i)} \left(1 - G_{x,j}\right). \tag{20}$$

For the interference caused by detection errors, when detecting the channel F_j, two detection errors may occur, namely, false detection and missed detection. Use H_1 to indicate that the channel is occupied by high-priority services, H_0 to indicate that there is no service in the channel, and P_e to indicate the probability of error detection, that is, $P_e(H_1 \mid H_0)$. This probability indicates that the high-priority service does not appear, but a detection error has occurred, and the cognitive node considers that the high-priority service appears and therefore causes the probability that the low-priority service in the channel is discarded or switched to other available channels for transmission. P_l represents the probability of missed detection, that is, $P_l(H_0 \mid H_1)$. This probability indicates that missed detection occurred when high-priority services appeared, and the cognitive node did not process the low-priority services transmitted in the channel, resulting in high-priority services and low-priority services, the probability of a collision. Therefore, for channel F_j, the probability of detection error is

$$E_j = P_e(H_1 \mid H_0) + P_l(H_0 \mid H_1). \tag{21}$$

When a service loses its current channel, the system will allocate another available channel to the user. This will cause

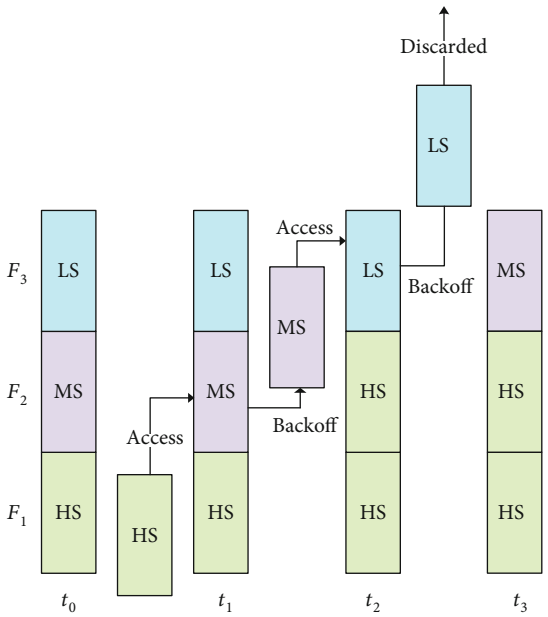


Figure 3: Priority-based channel scheduling strategy.

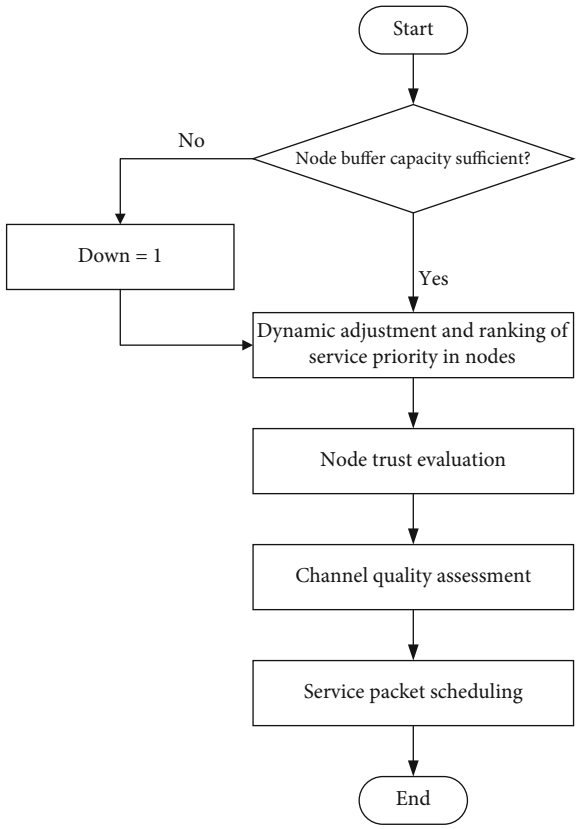


Figure 4: Flow chart of power wireless communication service packet scheduling mechanism based on dynamic priority.

interference from the switched user to the original user in the new channel. Therefore, T_j can be used to represent the interference caused during channel switching.

Table 3: Simulation parameter setting.

Simulation parameters	Parameter value setting
Number of nodes	100
Simulation time (simulation)	24 h
Message TTL	30 min
Node minimum speed	0.5 m/s
Node maximum rate	1 m/s
Node communication range	20 m
Minimum number of copies L	8
Initial constant P_{ini}	0.75
Decay weight γ	0.97
Transfer weight β	0.25
Message size range	[100, 300] kb

The available channel resource information can be obtained from the information exchange between neighboring nodes. For node i, the reliability of channel F_j can be obtained by calculating two types of interference probabilities:

$$K_{i,j} = \left(1 - G_{i,j}\right)\left(1 - T_j\right). \tag{22}$$

(3) Channel stability: stability refers to the ratio of the time when there is no service in the channel and the total time within a period of time. The channel state can be modeled as an ON-OFF model, which is an alternate update process. ON (occupied by business) and OFF (idle) duration obey the exponential distribution of parameters λ and μ, respectively [36], which are represented by T_{ON} and T_{OFF} random variables. In the model, the process of business from ON to OFF to ON in the channel is regarded as a cycle; then, the stability W_j of channel F_j is the update period length from ON to OFF to ON and the reference value of the relative stable update period length, the ratio of R. W_j can be expressed as

$$W_j = \frac{E(T_{\text{ON}}) + E(T_{\text{OFF}})}{R_j} = \frac{1/\mu_j^{\text{PU}} + 1/\lambda_j^{\text{PU}}}{R_j} = \frac{\mu_j^{\text{PU}} + \lambda_j^{\text{PU}}}{\mu_j^{\text{PU}}\lambda_j^{\text{PU}}R_j}, \tag{23}$$

where T_{ON} is the length of time that the channel is occupied by the service, T_{OFF} is the length of time that the channel does not have a service, μ_j^{PU} is the number of times the channel F_j is occupied by the service in a unit time, and λ_j^{PU} is the number of times the channel is idle in a unit time.

In summary, for node i, the channel quality parameter $V_{i,j}$ of channel F_j can be expressed as

$$\begin{aligned} V_{i,j} &= L_{i,j}\left[\gamma K_{i,j} + (1\text{-}\gamma) W_j\right] \\ &= L_{i,j}\left[\gamma\left(1 - Z_j\right)\left(1 - G_{i,j}\right)\left(1 - E_j\right)\left(1 - T_j\right) + (1\text{-}\gamma) W_j\right], \end{aligned} \tag{24}$$

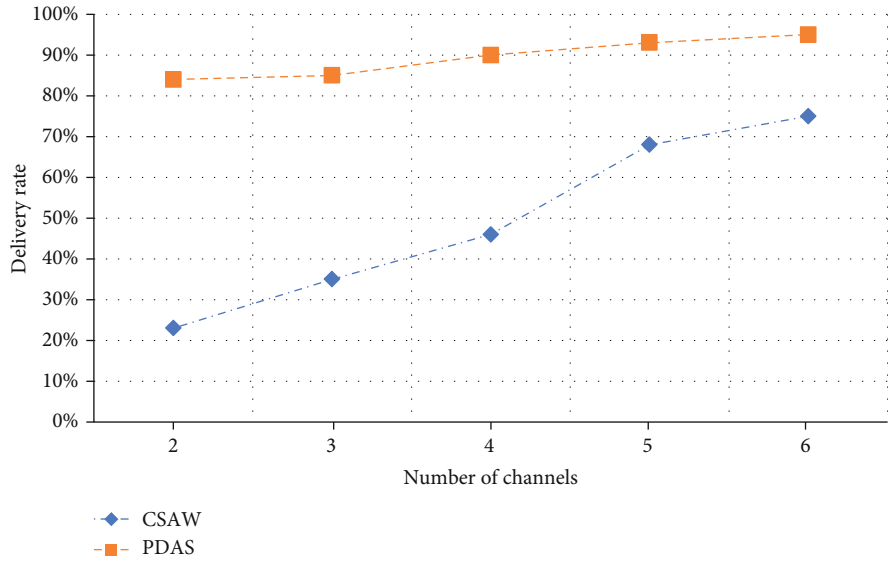


Figure 5: Delivery rate of the H service.

where γ is the weighting coefficient, which can be adjusted according to the network characteristics and service requirements and at the same time can reflect the network's emphasis on reliability and channel stability.

4.4. Packet Scheduling Mechanism. Due to the different access channel capabilities of different priority services, the traffic blocking situation is also different [37]. Here, we can divide into three kinds of services according to the service priority: high-priority service (HS), medium-priority service (MS), and low-priority service (LS). For LS, when all the channels are occupied by HS and MS packets, blocking occurs. It should be noted that the packets here contain packets (HS) whose priority is raised to a higher priority through a dynamic priority adjustment strategy. Each service uses idle spectrum resources to transmit data packets. If high-priority service appears again in the information transmission process of low-priority service, the low-priority service should be discarded from the channel or switched to other channels to continue transmission. In addition, the channel occupied by lower priority traffic should be selected as far as possible to prevent frequent handoff. The specific channel access, backoff, and handover strategies are shown in Figure 3.

As shown in Figure 3, at t_0 time, each service occupies the free frequency bands F_1, F_2, and F_3 for data transmission; at t_1 time, the authorized high-priority service (HS) access of F_2 forces the interruption of medium-priority service (MS) communication on F_2. MS backoff and access channel are occupied by the lowest priority service LS. At this time, the least priority LS data in the system is discarded. After a negligible delay time, at t_3 time, the system is in a stable state.

To properly schedule the packets, we select the optimal channel allocation decision through genetic algorithm (GA) and encode the entire channel allocation matrix by the minimum interval coding scheme. An individual in GA corresponds to a channel allocation scheme; the individual with the highest fitness in each generation population (channel allocation matrix) is taken as the optimal solution of the system.

4.5. Overall Flow of Algorithm. In this paper, the power communication service packet scheduling mechanism based on dynamic priority is proposed, which is divided into four steps: the dynamic adjustment strategy of service priority, evaluation of node trust value, channel quality assessment, and the scheduling of service packet, thus forming a complete power communication service data transmission process.

In the first part, the dynamic adjustment strategy of service priority determines the priority queue of the service in the node by comprehensively considering the TTL of the message in the node, evaluating the number and priority of the current copies of the service packet.

In the second part, we introduce the trust value evaluation method of wireless forwarding nodes to improve the security of data transmission.

In the third part, the channel quality is comprehensively evaluated from the three perspectives of channel connectivity, reliability, and stability.

In the fourth part, a flexible and efficient packet scheduling mechanism is designed for different priority services to allocate communication channels efficiently and improve the utilization of the system.

The algorithm is shown in Figure 4.

5. Simulation

This project is based on the simulation platform ONE and establishes the experimental simulation of the network. The node's movement model is a random walk model. The specific parameter settings of the node are shown in Table 3.

In the experiment, according to the different requirements of different services for network performance, three types of services are defined: emergency service, data flow service, and best effort service. It is defined as H class, the target delay is 800-1200 s; M class, the target delay is 1200-1800 s; and L class, the target delay is 1800-3000 s. The ratio of the number of messages generated by the system in the

network simulation is 1 : 3 : 6. The algorithm proposed in this paper is abbreviated as PDAS. The experimental comparison algorithms include priority insensitive spray wait routing with congestion control (CSAW) and priority aware routing QoS policy [25]. QoS policy adapts to resource allocation in a dynamic environment and defines constraints that optimize network-wide performance while satisfying the QoS constraints of a single category. However, the contagion strategy it uses consumes a lot of resources, and scheduling information needs to be provided globally, which is not suitable for delay-tolerant networks. Practical application.

5.1. Simulation Experiment Index. In this paper, the performance of the network is measured by message delivery rate, average delay, packet loss rate, and network overhead. The calculation methods of these four indicators are slightly different from those of traditional network service quality-related indicators.

(1) Delivery rate: the ratio of the total number of messages successfully delivered to the destination node to the total number of messages generated by the source node. A copy of the same message counts as one message

$$\text{Delivery rate} = \frac{\text{the total number of messages successfully delivered to the destination node}}{\text{the total number of messages generated by the source node}} \tag{25}$$

(2) Packet loss rate: the ratio of the total number of dropped packets to the total number of messages successfully delivered to the destination node. Different copies of the same message count as multiple packets. The higher the packet loss rate, the worse the network performance

$$\text{Packet loss rate} = \frac{\text{total number of dropped messages}}{\text{the total number of messages received}} \tag{26}$$

(3) Network overhead: it is determined by the total number of forwarding messages and the total number of messages successfully delivered to the destination node. Copies of all messages included in the total number of forwards of all messages

$$\text{Network overhead} = \frac{\text{the total number of forwarding of all messages-the total number of messages successfully delivered to the destination node}}{\text{the total number of messages successfully delivered to the destination node}} \tag{27}$$

5.2. Simulation Results. The simulation mainly verifies the performance of the algorithm under different network resources. When the cache space in the network is insufficient, it will cause network congestion and affect the quality of service. Therefore, the simulation experiment mainly compares the routing conditions in different buffer spaces to compare the performance of routing algorithms.

It can be seen from Figure 5 that using PDAS can effectively improve the delivery rate of class H services. In addition, even when the network load is heavy, the delivery rate can be maintained at about 90%. This is the highest priority service in the channel. It can seize the channel occupied by other services and get more transmission opportunities.

As can be seen from Figure 6 after the adoption of PDAS, the delivery rate of class M and class L services has also improved. This is because when the network load is heavy, with the increase of queuing delay, the priority level of the packets to be transmitted is increased according to the priority dynamic adjustment strategy, so as to obtain more transmission resources.

Figure 7 shows the relationship between the packet loss rate and the number of channels. It can be seen that when the network resources are sufficient, the packet loss rates of all algorithms are gradually reduced. Because QoS policy algorithm uses infection algorithm, the redundancy of service messages in the network will still be higher than that of the spray wait routing algorithm. Therefore, even in the case of sufficient network resources, the packet loss rate of the QoS policy algorithm for low-priority services is still relatively high.

Figure 8 shows the relationship between network overhead and channel number. It can be seen that in the case of lack of network resources, the cost of the whole network is relatively large. With the increase of the number of channels, the network overhead of the PDAS algorithm proposed in this paper is low. In the PDAS algorithm, the priority of traffic and channel resources will be dynamically adjusted before entering the route, which can reduce the redundant forwarding times and reduce the network overhead. Because the QoS policy algorithm uses the infection algorithm, it is difficult to

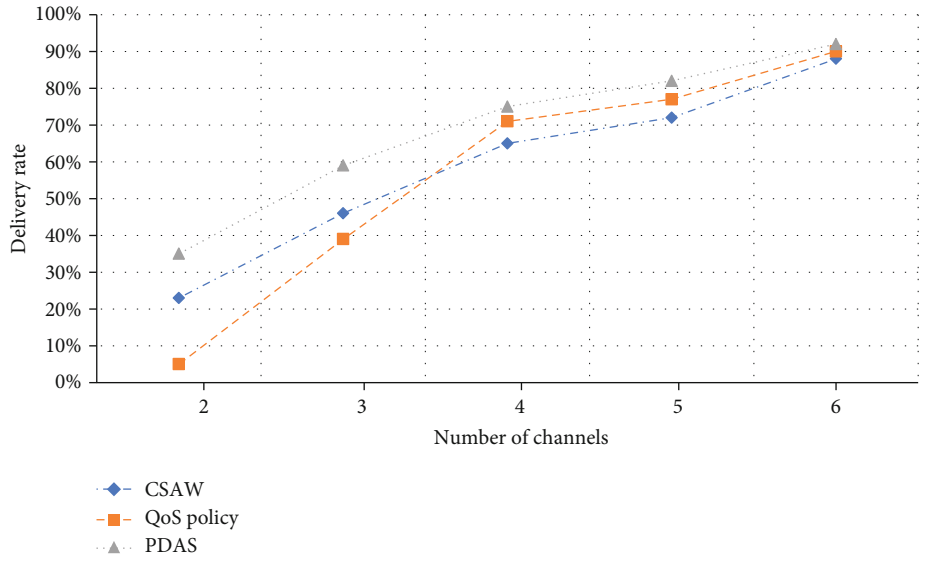


Figure 6: Delivery rate of the M service and L service.

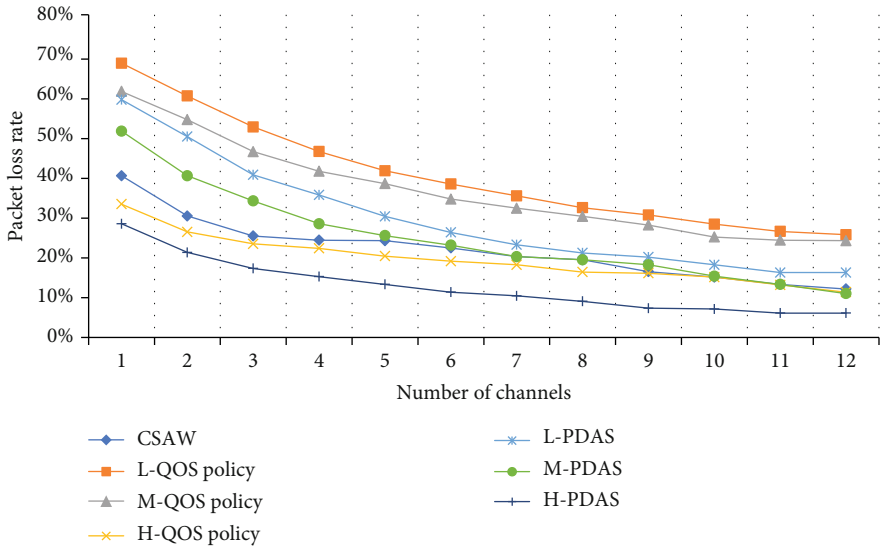


Figure 7: Packet loss rate of the algorithm.

control the redundancy of service messages in the network, but excessive redundancy will increase the number of messages to be forwarded, which will lead to larger network overhead. Although the CSAW algorithm limits the number of copies of service messages, it does not adjust the channel, so there will be unnecessary forwarding, resulting in increased network overhead.

6. Conclusions

In order to provide reliable QoS guarantee for different services under the real-time change of spectrum resources in power communication network, this paper proposes a communication service scheduling method based on dynamic priority. This method solves the problem that the traditional scheduling mechanism only considers the absolute priority of services and ignores the relative priority among services, which cannot meet the requirement of intelligent power communication network to provide differentiated QoS services for heterogeneous services. The simulation results show that by using the proposed method, the system can ensure the communication performance of services with high QoS requirements without interference and improve the utilization of the whole system. The power wireless communication network scenarios involved in this article have a certain degree of promotion and provide theoretical support for ensuring the QoS for power communication services. However, there are still some differences between the scenario

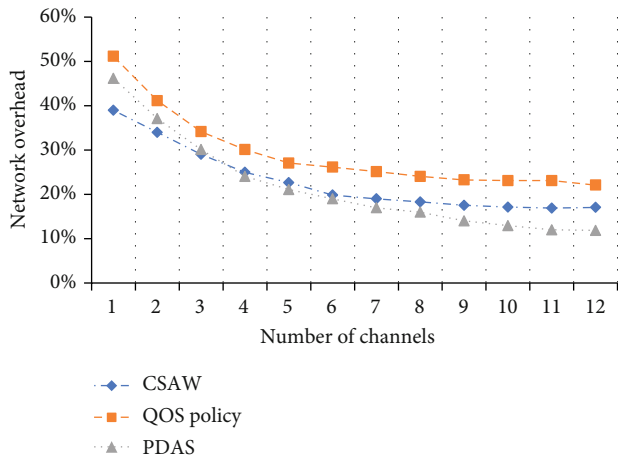


Figure 8: Network overhead of the algorithm.

set in this article and the actual network. Besides, we will consider channel multiplexing to realize simultaneous data transmission of multiple services in future research.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Science and Technology Project of State Grid Corporation of China: Research and Application of Key Technologies in Virtual Operation of Information and Communication Resources.

References

[1] G. A. Shah, V. C. Gungor, and O. B. Akan, "A cross-layer design for QoS support in cognitive radio sensor networks for smart grid applications," in *2012 IEEE International Conference on Communications (ICC)*pp. 1378–1382, Ottawa, Canada, Ottawa, ON, Canada, 2012.

[2] G. A. Shah, V. C. Gungor, and O. B. Akan, "A cross-layer QoS-aware communication framework in cognitive radio sensor networks for smart grid applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1477–1485, 2013.

[3] J. G. Deshpande, E. Kim, and M. Thottan, "Differentiated services QoS in smart grid communication networks," *Bell Labs Technical Journal*, vol. 16, no. 3, pp. 61–81, 2011.

[4] IEC 61850-5, *Communication networks and systems in substations – Communication requirements for functions and device models*, IEC International Standard, 2 edition, 2013.

[5] J. Guo, "Study on the structure of electric power communication network of strong and smart grid in China," in *2010 International Conference on Power System Technology*, pp. 1–3, Hangzhou, China, 2010.

[6] R. Yu, W. Zhong, S. Xie, Y. Zhang, and Y. Zhang, "QoS differential scheduling in cognitive-radio-based smart grid networks: an adaptive dynamic programming approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 435–443, 2016.

[7] A. Roy, S. Mahanta, M. Tripathy, S. Ghosh, and S. Bal, "Health condition identification of affected people in post disaster area using DTN," in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 1–3, New York, NY, USA, 2016.

[8] S. Sato, M. Takai, Y. Owada, T. Maeno, and M. Oguchi, "Development of a request processing method for relief goods in the distributed material management support system of evacuation shelters using DTN," in *2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 94–98, Lagos, 2019.

[9] N. Uchida, T. Shingai, T. Shigetome, and Y. Shibata, "Implementations of data triage methods for DTN based disaster information networks," in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*, pp. 205–209, Taichung, 2017.

[10] N. Ruangchaijatupon and J. Yu-sheng, "Simple proportional fairness scheduling for OFDMA frame-based wireless systems," in *IEEE Wireless Communications and Networking Conference*, pp. 1593–1597, Las Vegas, NV, USA, 2008.

[11] "IEEE standard communication delivery time performance requirements for electric power substation automation," *IEEE Std 1646-2004*, pp. 1–36, 2005.

[12] M. J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *2011 Proceedings IEEE INFOCOM*, pp. 1728–1736, Shanghai, 2011.

[13] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications*, vol. 7, no. 3, pp. 239–254, 2003.

[14] S. Tajima, T. Asaka, and T. Takahashi, "Priority control using multi-buffer for DTN," in *The 16th Asia-Pacific Network Operations and Management Symposium*, pp. 1–6, Hsinchu, 2014.

[15] Z. Du, C. Wu, T. Yoshinaga, and Y. Ji, "A prophet-based DTN protocol for VANETs," in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 1876–1879, Guangzhou, 2018.

[16] B. Huang, L. Liu, H. Zhang, Y. Li, and Q. Sun, "Distributed Optimal Economic Dispatch for Microgrids Considering Communication Delays," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 8, pp. 1634–1642, 2019.

[17] A. J. Mashhadi and L. Capra, "Priority scheduling for participatory delay tolerant networks," in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–3, Lucca, Italy, 2011.

[18] Z. Gao, Y. Li, and S. Wan, "Exploring deep learning for view-based 3D model retrieval," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1, pp. 1–21, 2020.

[19] Y. Xi, Y. Zhang, S. Ding, and S. Wan, "Visual question answering model based on visual relationship detection," *Signal Processing: Image Communication*, vol. 80, article 115648, 2020.

[20] A. Zhou, S. Wang, S. Wan, and L. Qi, "LMM: latency-aware micro-service mashup in mobile edge computing environment," *Neural Computing and Applications*, vol. 32, no. 19, pp. 15411–15425, 2020.

[21] X. Xu, Q. Wu, L. Qi, W. Dou, S.-B. Tsai, and M. Z. A. Bhuiyan, "Trust-aware service offloading for video surveillance in edge computing enabled Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[22] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[23] C. Zhang, X. Guo, X. Guo et al., "Machine learning model comparison for automatic segmentation of intracoronary optical coherence tomography and plaque cap thickness quantification," *Computer Modeling in Engineering & Sciences*, vol. 123, no. 2, pp. 631–646, 2020.

[24] T. Ma, H. Zhou, D. Jia, A. al-Dhelaan, M. al-Dhelaan, and Y. Tian, "Feature selection with a local search strategy based on the forest optimization algorithm," *Computer Modeling in Engineering & Sciences*, vol. 121, no. 2, pp. 569–592, 2019.

[25] P. Matzakos, T. Spyropoulos, and C. Bonnet, "Joint scheduling and buffer management policies for DTN applications of different traffic classes," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2818–2834, 2018.

[26] S. Wan, X. Xu, T. Wang, and Z. Gu, "An intelligent video analysis method for abnormal event detection in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2020.

[27] 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.

[28] B. Huang, L. Liu, Y. Li, and H. Zhang, "Distributed optimal energy management for microgrids in the presence of time-varying communication delays," *IEEE Access*, vol. 7, pp. 83702–83712, 2019.

[29] 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*, vol. 32, no. 9, pp. 4387–4415, 2020.

[30] E. Wang, Y. Yang, and J. Wu, "A Buffer Management Strategy on Spray and Wait Routing Protocol in DTNs," in *2015 44th International Conference on Parallel Processing*, pp. 799–808, Beijing, China, 2015.

[31] G. Han, J. Jiang, L. Shu, and M. Guizani, "An attack-resistant trust model based on multidimensional trust metrics in underwater acoustic sensor network," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2447–2459, 2015.

[32] Q. Liu, Y. Liao, B. Tang, and L. Yu, "A trust model based on subjective logic for multi-domains in grids," in *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pp. 882–886, Wuhan, 2008.

[33] J. Feng, F. Richard Yu, Q. Pei, X. Chu, J. du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6214–6228, 2020.

[34] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[35] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y.-J. Song, "Group-based trust management scheme for clustered wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1698–1712, 2009.

[36] Hsien-Po Shiang and M. van der Schaar, "Queuing-based dynamic channel selection for heterogeneous multimedia applications over cognitive radio networks," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 896–909, 2008.

[37] R. Yu, C. Zhang, X. Zhang, L. Zhou, and K. Yang, "Hybrid spectrum access in cognitive-radio-based smart-grid communications systems," *IEEE Systems Journal*, vol. 8, no. 2, pp. 577–587, 2014.