

Research Article

The Improved Nonparametric Regression Model for the IoT Link Load Balancing Control Algorithm

Qingping Xue 

Henan University of Animal Husbandry and Economy, Zhengzhou 450044, China

Correspondence should be addressed to Qingping Xue; 81017@hnuah.edu.cn

Received 22 December 2021; Revised 25 February 2022; Accepted 17 March 2022; Published 4 April 2022

Academic Editor: Mu Zhou

Copyright © 2022 Qingping Xue. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to improve the load balance control effect of the Internet of Things link, this paper combines the nonparametric regression model to improve the load balancing algorithm of the Internet of Things link. Moreover, this paper proposes a load balancing research strategy based on data plane data flow, which is aimed at improving the load balancing problem of data flow in the link. The data center network uses a multilayer fat tree topology to store information in the flow table corresponding to the switch for data flow processing and forwarding operations. In addition, this paper constructs a load balancing model for intelligent Internet of Things link and verifies the model in this paper through experimental research. The research results show that the load balancing control algorithm for Internet of Things link based on the nonparametric regression model proposed in this paper can effectively improve the internal scheduling of the Internet of Things link system and promote the load balancing effect.

1. Introduction

With the in-depth study of WSN by scientific researchers, they have found that it has a wide range of applications but also has shortcomings, such as sensor node energy, communication distance, computing, and storage capabilities that are very limited. Since each sensor node is powered by a battery with limited energy, the complicated deployment environment brings a lot of inconvenience to the battery replacement work [1]. In addition, with the exhaustion of sensor node energy and the addition of new sensor nodes, the topology of the network will change. In the traditional network hierarchy, only two adjacent layers can communicate with each other, and it is difficult to share information with each other between two nonadjacent layers. Therefore, there are great limitations, which can no longer be well adapted to the development needs of WSN [2]. How to design a good performance energy optimization algorithm to meet the high requirements of WSN on the network life cycle has become the research focus and difficulty of WSN in recent years [3].

High-rate wireless sensor network (HRWSN) integrates information perception, processing, and transmission and

is widely used in military, industrial and agricultural control, biomedicine, environmental monitoring, disaster relief, and other fields. It is a key technical link from sensor networks to the Internet of Things and pervasive computing. In order to meet high-quality video surveillance, diversified information collection, complex task processing, high-precision positioning, and other applications, it is urgent to introduce information-rich images, audio, and video into high-rate sensor networks. High-rate sensor networks have the characteristics of rich sensing information, complex processing tasks, node movement, dynamic changes in topology, and severe energy limitation.

This paper combines the nonparametric regression model to improve the Internet of Things link load balancing algorithm, builds an intelligent model, and verifies the effectiveness of this model through research.

2. Related Work

Firstly, the research and practical application of WSN related theories are reviewed. Researchers have actively carried out research and practical application of WSN related theories and have put forward energy optimization schemes

for sensor nodes one after another. Literature [4] proposed the LEACH (Low Energy Adaptive Clustering Hierarchy) algorithm, a typical representative of clustering routing algorithms. Through periodic random selection of cluster heads, each node has an equal probability of being a cluster head, and the network energy consumption is balanced to the entire network; [5] uses simulated annealing algorithm to divide the clusters, and the remaining energy of the nodes is considered when selecting the cluster heads; the number of times the nodes serve as cluster heads is considered in the literature [6]; the pLEACH algorithm proposed in the literature [7] divides the network, and the node with the highest remaining energy is selected as the cluster head in each subarea; literature [8] balances energy consumption by adjusting the data transmission rate; literature [9] improves the selection method of cluster heads and determines the remaining energy of nodes and neighbor nodes. The number and location information are considered; literature [10] uses fuzzy theory to calculate the probability of a node as a cluster head to improve the threshold formula; the LEACH-EC algorithm proposed in literature [11] optimizes the selection of cluster heads through the remaining energy and connectivity of the node. In recent years, some new intelligent algorithms have also been proposed. Literature [12] uses particle swarm optimization (PSO) to optimize the selection of cluster heads; literature [13] uses the ant colony algorithm to optimize data transmission. Path. Literature [14] based on the cell membrane optimization algorithm to complete the clustering of nodes through the three factors of concentration, energy, and distance; in addition, some researchers have proposed new energy supply methods, but they are limited by the cost of WSN, and it is difficult to realize the consideration of volume. In literature [15], wireless charging is used for the sensor node to deal with the energy hole problem, and the literature [16] uses a smart solar energy collection system to provide stable power supply to the sensor node.

The data exchange process involves multiple sets of data fusion and data processing, which puts forward certain requirements for load balancing. The related work is analyzed below. The traditional data center network is generally a tree topology with two or three layers of switching structure. Although it can accommodate tens of thousands of servers, with the rapid expansion of the data center scale and the continuous influx of various new services, there is a tree-based data center. The structure shows the shortcomings of low utilization of network resources and poor scalability; so, the data center network topology has begun to transform to a new type of network structure. The structure of literature [17] has become a new and popular network topology with its simple structure, easy deployment, and the advantages of non-blocking transmission and half-bandwidth. According to its characteristics, the fat-tree structure generally adopts ECMP (equal cost multipath-based) or its improved technology to realize traffic forwarding. ECMP, as its name implies, uses multiple equivalent paths to forward traffic. It uses the diversity of the underlying path. When a packet arrives, it selects a path by modulo the number of paths obtained by the hash value of the

header of the packet. In this way, the flow will be forwarded along the same path. Since the distribution probability of the value after the header of each flow is equalized, each path has an equal utilization rate, and to a certain extent, the flow can be balanced to achieve the effect of load balancing. However, ECMP also has its own characteristics and limitations. One of the most critical issues is that although most of the data center network traffic is small, a small amount of continuous large flow (we will define it as a stable flow later) accounts for the vast majority. For most bandwidth [18], once two or more long-lived stable flows choose the same egress for forwarding, congestion will occur, resulting in unbalanced distribution of link traffic, and seriously affecting network transmission performance. The test results in literature [19]. It means that the transmission performance can be reduced by 50% in this case. If it is just to balance the link traffic, you can also use packet-based polling and forwarding. This method forwards each message to each outlet in turn according to the FIFO principle, almost ensuring that the number of packets forwarded by each path is the same, but even if it is, the improved balance polling algorithm [20] also cannot avoid a problem; that is, packets arrive out of order. From the perspective of TCP flow, the out-of-sequence arrival of packets is regarded as link congestion, which reduces the sending window and significantly degrades TCP performance. In contrast, ECMP is more suitable for the fat-tree structure. How to implement a dynamic flow routing and scheduling to solve the congestion problem caused by stable flow aggregation in ECMP has become the focus of this paper. For traditional networks, data center network load balancing is performed on each network node, and it is impossible to obtain all node attributes and link information in real time during network operation. It is very difficult to achieve the abovementioned dynamic load balancing effect. But fortunately, with the emergence of SDN technology, the controller can communicate with switches that support the OpenFlow protocol through the OpenFlow protocol [21] to obtain real-time statistical information that can truly and effectively reflect the network status. Based on these statistical information, we can perform calculations more accurately and efficiently from a global perspective, which also makes it possible to implement the abovementioned dynamic load balancing algorithm.

3. Nonparametric Regression Model and Local Polynomial Regression Estimation

For n observations $(x_1, Y_1), \dots, (x_n, Y_n)$, the relationship between the response variable Y and the covariate x is defined by the following equation:

$$Y_i = r(x_i) + \varepsilon_i, \quad E(\varepsilon_i) = 0, \quad i = 1, 2, \dots, n. \quad (1)$$

Among them, r is the regression function, the variable x is also called feature, and the estimate of $r(x)$ is denoted by $\hat{r}_n(x)$.

Under the conditions of Nadaraya-Watson kernel estimation, we consider choosing an estimator $a \equiv \hat{r}_n(x)$ to

minimize the sum of squares $\sum_{i=1}^n (Y_i - a)^2$. We define the weight function $w_i(x) = K((x_i - x)/h)$ and choose $a \equiv \hat{r}_n(x)$ to minimize the following weighted sum of squares:

$$\sum_{i=1}^n w_i(x)(Y_i - a)^2. \quad (2)$$

The solution is

$$\hat{r}_n(x) \equiv \frac{\sum_{i=1}^n w_i(x)Y_i}{\sum_{i=1}^n w_i(x)}. \quad (3)$$

It happens to be a nuclear regression estimate.

We use a local polynomial of p order to improve the estimation. For a value u in a neighborhood of x , the polynomial is

$$P_x(u; a) = a_0 + a_1(u - x) + \frac{a_2}{2!}(u - x)^2 + \dots + \frac{a_p}{p!}(u - x)^p. \quad (4)$$

In a neighborhood of x , we use the following polynomial to approximate a smooth regression function $r(u)$:

$$r(u) \approx P_x(u; a). \quad (5)$$

We choose $\hat{a} = (\hat{a}_0, \dots, \hat{a}_p)^T$ that minimizes the following local weighted sum of squares to estimate $a = (a_0, \dots, a_p)^T$:

$$\sum_{i=1}^n w_i(x)[Y_i - P_x(X_i; a)]^2. \quad (6)$$

It is estimated that \hat{a} depends on x . $\hat{a}(x) = (\hat{a}_0(x), \dots, \hat{a}_p(x))^T$, and the estimate of $r(\cdot)$ is $\hat{r}_n(u) = P_x(u; \hat{a})$. We set the following:

$$X_x = \begin{bmatrix} 1 & x_1 - x & \dots & \frac{(x_1 - x)^p}{p!} \\ 1 & x_2 - x & \dots & \frac{(x_2 - x)^p}{p!} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n - x & \dots & \frac{(x_n - x)^p}{p!} \end{bmatrix}. \quad (7)$$

W_x is the diagonal matrix of $n \times n$, where $w_{ij} = w_i(x)$. Formula (6) is written as

$$(Y - X_x a)^T W_x (Y - X_x a). \quad (8)$$

By minimizing formula (8), the weighted least squares estimate is obtained:

$$\hat{a}(x) = (X_x^T W_x X_x)^{-1} X_x^T W_x Y. \quad (9)$$

$\hat{r}_n(x) = \hat{a}_0(x)$ is the inner product of the first row of $(X_x^T W_x X_x)^{-1} X_x^T W_x$ and Y .

The local polynomial regression is estimated as

$$\hat{r}_n(x) = \sum_{i=1}^n l_i(x) Y_i. \quad (10)$$

Among them,

$$L(x)^T = (l_1(x), \dots, l_n(x)), \quad L(x) = e_1^T (X_x^T W_x X_x)^{-1} X_x^T W_x. \quad (11)$$

The mean and variance of this estimate are

$$\begin{aligned} E(\hat{r}_n(x)) &= \sum_{i=1}^n l_i(x) r(x_i), \\ \text{Var}(\hat{r}_n(x)) &= \sigma^2 \sum_{i=1}^n l_i(x)^2 = \sigma^2 \|L(x)\|^2. \end{aligned} \quad (12)$$

The risk (mean square error) is

$$R(h) = E\left(\frac{1}{n} \sum_{i=1}^n [\hat{r}_n(x_i) - r(x_i)]^2\right). \quad (13)$$

We need to choose the smoothing parameter h . Ideally, we want to choose the $R(h)$ that minimizes ν . However, $R(h)$ depends on the unknown function $r(x)$. Instead, the estimated $\hat{R}(h)$ of $R(h)$ will be minimized. We use the missing crossvalidation score defined below to estimate risk.

The missing crossvalidation score is defined as

$$CV = \hat{R}(h) = \frac{1}{n} \sum_{i=1}^n [Y_i - \hat{r}_{(-i)}(x_i)]^2. \quad (14)$$

Among them, $\hat{r}_{(-i)}$ is the estimate obtained from the i -th data point (x_i, Y_i) .

$\hat{r}_{(-i)}$ is defined as

$$\hat{r}_{(-i)}(x) = \sum_{j=1}^n Y_j \ell_{j,(-i)}(x). \quad (15)$$

Among them,

$$\ell_{j,(-i)}(x) = \begin{cases} 0, & j = i, \\ \frac{\ell_j(x)}{\sum_{k \neq i} \ell_k(x)}, & j \neq i. \end{cases} \quad (16)$$

In other words, the weight of x_i is 0, and the other weights are reregularized so that their sum is 1.

The intuitive meaning of crossvalidation is due to

$$\begin{aligned} E\left(Y_i - \hat{r}_{(-i)}(x_i)\right)^2 &= E\left(Y_i - r(x_i) + r(x_i) - \hat{r}_{(-i)}(x_i)\right)^2 \\ &= \sigma^2 + E\left(r(x_i) - \hat{r}_{(-i)}(x_i)\right)^2 \approx \sigma^2 + E\left(r(x_i) - \hat{r}_n(x_i)\right)^2. \end{aligned} \quad (17)$$

Predictive risk is equal to $R(r, \hat{r}_n) + \sigma^2$ [21], and then

$$E(\hat{R}) \approx R + \sigma^2. \quad (18)$$

$E(\hat{R})$ equals predictive risk. In this way, the cross-validation score is an almost unbiased estimate of risk.

For linear smoothers, there is a simple formula for calculating $\hat{R}(h)$.

\hat{r}_n is a linear smoother; then, the missing crossvalidation score $\hat{R}(h)$ can be written as

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^n \left[\frac{Y_i - \hat{r}_n(x_i)}{1 - L_{ii}} \right]. \quad (19)$$

Among them, $L_{ii} = \ell_i(x_i)$ is the i -th diagonal element of the smooth matrix L .

The smoothing parameter h can be selected by minimizing the crossvalidation score $\hat{R}(h)$.

If we do not minimize the crossvalidation score, we can use another approximation method called generalized cross-validation. Here, we replace each L_{ii} with its average $n^{-1} \sum_{i=1}^n L_{ii} = V/n$, and $V = \text{tr}(L)$ is the effective degree of freedom. This will minimize the following equation:

$$\text{GCV}(h) = \frac{1}{n} \sum_{i=1}^n \left[\frac{Y_i - \hat{r}_n(x_i)}{1 - v/n} \right]^2. \quad (20)$$

Generally, the bandwidth that minimizes the generalized crossvalidation score is close to the bandwidth that minimizes the crossvalidation score.

By using approximate $(1 - x)^{-1} \approx 1 + 2x$, we can obtain

$$\text{GCV}(h) \approx \frac{1}{n} \sum_{i=1}^n [Y_i - \hat{r}_n(x_i)]^2 + \frac{2v\hat{\sigma}^2}{n} \equiv C_p. \quad (21)$$

Among them, $\hat{\sigma}^2 = 1/n \sum_{i=1}^n [Y_i - \hat{r}_n(x_i)]^2$. Generally, for different choices of $\Theta(n, h)$, many common bandwidth selection criteria can be written as

$$B(h) = \Theta(n, h) \times \frac{1}{n} \sum_{i=1}^n [Y_i - \hat{r}_n(x_i)]^2, \quad (22)$$

$$L(\hat{h}) = \frac{1}{n} \sum_{i=1}^n [\hat{r}_n(x_i) - r(x_i)]^2.$$

Then, all \hat{h} , \hat{h}_0 and h_0 tend to 0 at the rate of $n^{-1/5}$. More-

over, for some normal numbers $C_1, C_2, \sigma_1, \sigma_2$, there are

$$\begin{aligned} n^{3/10}(\hat{h} - \hat{h}_0) &\longrightarrow N(0, \sigma_1^2) & n[L(\hat{h}) - L(\hat{h}_0)] &\longrightarrow C_1\chi_1^2, \\ n^{3/10}(h_0 - \hat{h}_0) &\longrightarrow N(0, \sigma_2^2) & n[L(h_0) - L(\hat{h}_0)] &\longrightarrow C_2\chi_1^2. \end{aligned} \quad (23)$$

In this way, the relative convergence rate of \hat{h} is

$$\frac{\hat{h} - \hat{h}_0}{\hat{h}_0} = \text{Op}\left(\frac{n^{3/10}}{n^{1/5}}\right) = \text{Op}(n^{-1/10}). \quad (24)$$

This slow convergence rate indicates that it is difficult to estimate the bandwidth. The convergence rate is essentially a question of bandwidth selection, because the following formula also holds:

$$\frac{\hat{h} - h_0}{h_0} = \text{Op}\left(\frac{n^{3/10}}{n^{1/5}}\right) = \text{Op}(n^{-1/10}). \quad (25)$$

4. Load Balancing Control Algorithm for Internet of Things Link Based on the Nonparametric Regression Model

The new SDN architecture puts forward the idea of completely separating the controller and the switch. The control plane realizes flexible control of data packets on the forwarding plane through custom programming, which provides a good platform for the innovative development of new networks and applications in the future. It connects the interaction between business applications, network services, and network devices through open interfaces, adopts a centralized control method, has more flexible software programmability, and can realize different network business logic. Among them, the network control layer provides an abstract network global view for the business application layer through the northbound interface API, so that the application layer can directly control the behavior of the network. It can also provide various application services for the network, mainly including network policies, security policies, QoS policies, and cloud services. The control layer and the underlying SDN switch realize data grouping and forwarding processing through the "control-forward" communication interface OpenFlow protocol. The biggest difference between an SDN network and a traditional computer network is that it can customize and flexibly manage the functions of network switching equipment through software programming. Among them, SDN has the three basic characteristics of separation of control and forwarding, open programmable interfaces, and logically centralized control, which brings great convenience to both network managers and application developers. The new SDN architecture is shown in Figure 1, which consists of four parts: an application plane, a control plane, a data plane, and a side management plane. Different interface protocols are used for communication

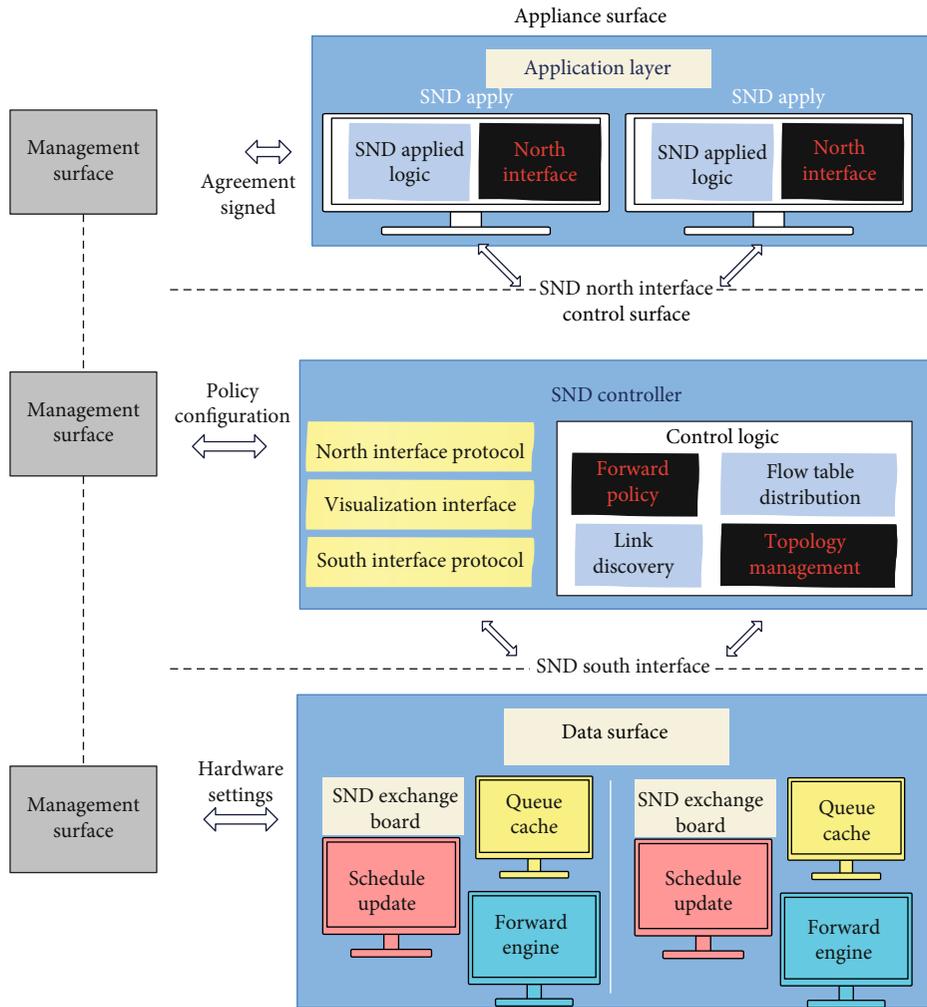


FIGURE 1: SDN architecture.

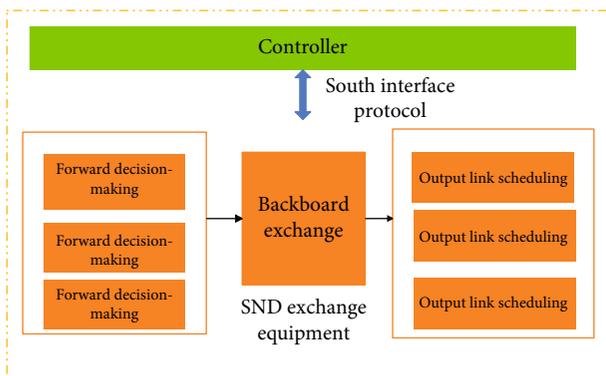


FIGURE 2: SDN switching equipment architecture.

and interconnection between the various planes, so that the entire architecture has better perception and control capabilities.

Different from the traditional network switching equipment, the SDN switching equipment architecture is shown in Figure 2. The controller is connected to the inherent

switching equipment of the SDN data plane through the southbound interface protocol to implement data forwarding. This decoupling architecture design reduces the complexity of network device design and improves the scalability of hardware switching devices.

The OpenFlow protocol is the most commonly used southbound interface protocol for connecting the data plane and the control plane in the SDN architecture. It is a universal, open, and vendor-independent interface protocol. In the SDN network architecture, the OpenFlow protocol is a communication interface used for information exchange between the controller and the switch. The schematic diagram of the OpenFlow protocol architecture is shown in Figure 3. It allows the controller in the control plane to access and manage the switching equipment of the data plane by issuing flow table rules. That is to say, when a newly arrived data packet reaches the OpenFlow switch, it first searches for a match in the flow table. If there is a matching item, it will directly perform the forwarding process. If there is no matching item, the OpenFlow switch will upload the data packet to the controller through the interactive

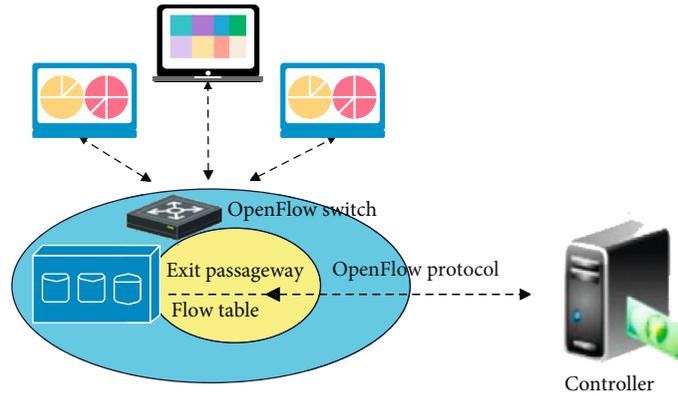


FIGURE 3: Schematic diagram of OpenFlow protocol architecture.

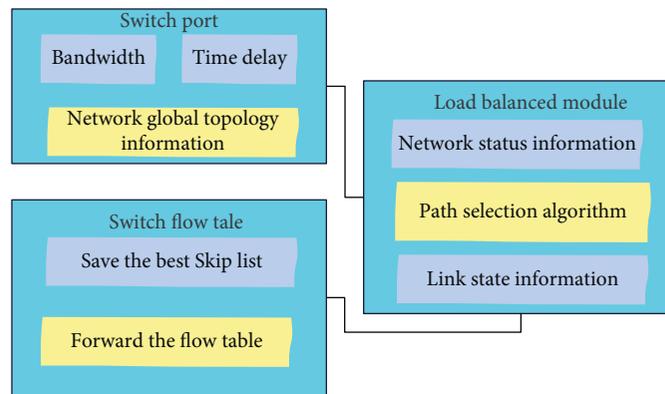


FIGURE 4: Overall architecture diagram of load balancing of data plane data flow.

communication interface OpenFlow protocol. Then, the controller uses routing calculation, forwarding decision, and other processing and then sends the forwarding mode of the data flow in the data packet to the OpenFlow switch through the OpenFlow protocol. Moreover, the forwarding mode is stored in the flow table of the switch for subsequent search and matching operations of the flow table.

According to the characteristics of data center network traffic that is relatively concentrated and burst, this paper proposes a load balancing research strategy based on data plane data flow, which is mainly aimed at the load balancing problem of data flow in the link. Data center networks usually adopt a multilayer fat tree topology. A top-level switch at the sending end can access the state information of the neighboring receiving-end switch or the bottom-level receiving-end switch through the next-level switch and save the information in the corresponding flow table of the switch for operations such as data flow processing and forwarding. The overall architecture diagram of the data flow processing of the switch data plane is shown in Figure 4, which includes a switch port module, a load balancing module, and a switch flow table module.

Under the premise of satisfying network connectivity and coverage, unnecessary communication links are eliminated through power control or backbone network node selection. The topology control algorithm is to simplify the densely deployed topology shown in Figure 5(a) to a simple

topology through power control or the selection of backbone network nodes, respectively, as shown in Figure 5(b) or Figure 5(c), so as to achieve the purpose of improving node energy utilization rate and prolonging the network lifetime. In Figure 5, circles represent nodes, black nodes are selected backbone network nodes, and dotted lines represent communication links between nodes.

The architecture diagram of the load balancing application is shown in Figure 6. Mininet implements the data center topology, Iperf simulates virtual machine communication, and the proxy&iperfcontroller script is used to collect the traffic matrix and maintain the iperf process. The database is used to record network throughput and link load information. The web is used to display network status information and algorithm control operations. The main logic of the load balancing application is implemented on the floodlight controller.

As can be seen in Figure 6, the load balancing application is mainly composed of four components: link load collection components, traffic collection components, load balancing algorithm components, and routing configuration components. The entire load balancing application has two asynchronous execution processes, namely, the flow analysis process and the link load analysis process.

Each user in the rate sensor network can include multiple types of services. The system bandwidth is B , the number of users in the active period is K , the number of subcarriers

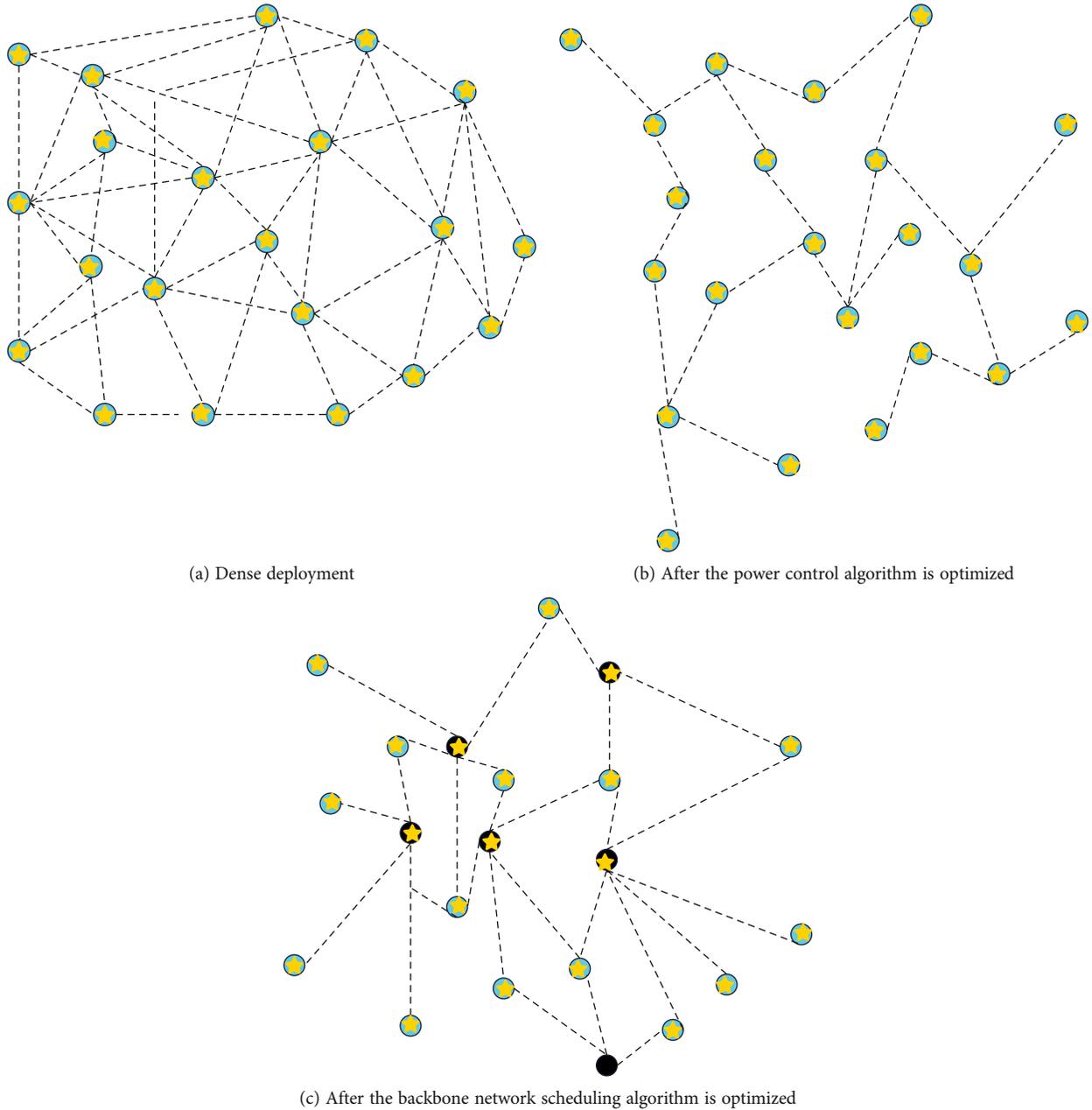


FIGURE 5: Topology control.

is N , and the bandwidth of each subcarrier is B/N . It is assumed that the system can obtain complete instantaneous state information of the channel and report the information to the base station in time through the error-free feedback channel. Assuming that N subcarriers are equally divided into n subchannels, the subchannels composed of subcarriers are almost flat fading channels. In order to improve spectrum utilization, adaptive modulation and coding (AMC) technology can be used on each subchannel. The system divides the system time in units of data frames, and each data frame time can be subdivided into S time slots. Assuming that the channel gain of each subchannel remains unchanged within each frame time, at the beginning of each frame, The base station estimates the state information CSI

of each subchannel in the link according to the information fed back by the user. According to the two-dimensional allocatable resources (subchannels, time slots) of the system, the shared channel is divided into multiple time-frequency units as the smallest unit of resource allocation; that is, a certain subchannel is occupied in frequency, and a time slot is occupied in time. $(1 < n < N)$ and $(1 < s < S)$, respectively, represent the subchannel sequence number and the sequence number within the data frame, so that there are a total of $N \times S$ block time-frequency units in a frame, and each time-frequency unit can only be one user occupied.

At the MAC layer, the base station allocates an independent data queue with limited capacity for each user, and the service order of the queue is first-come, first-served. Based

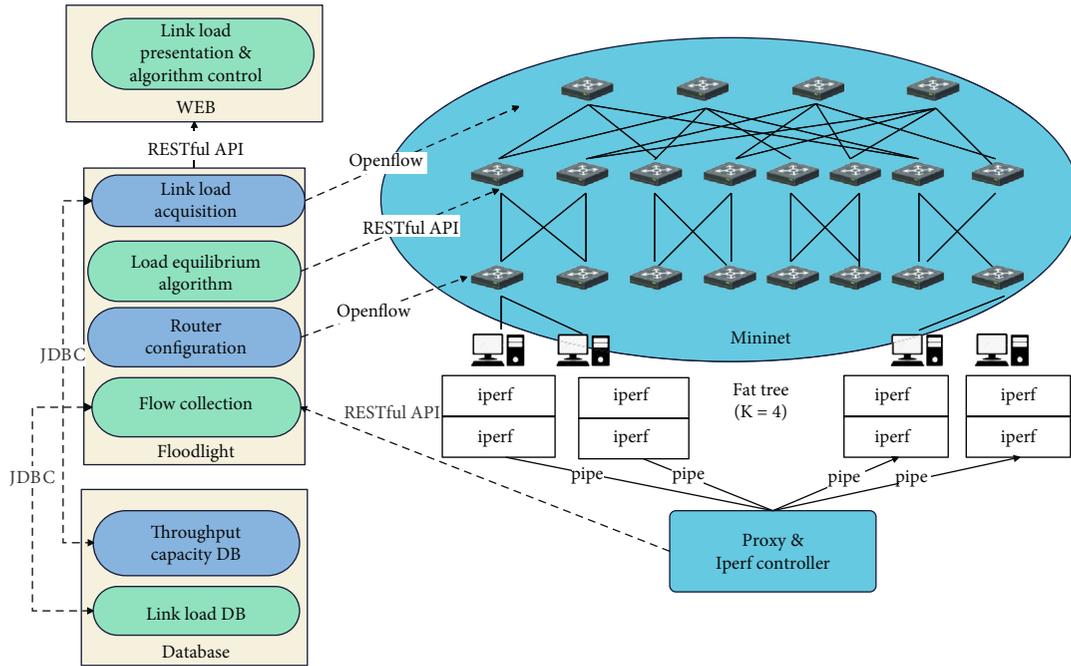


FIGURE 6: Load balancing application architecture diagram.

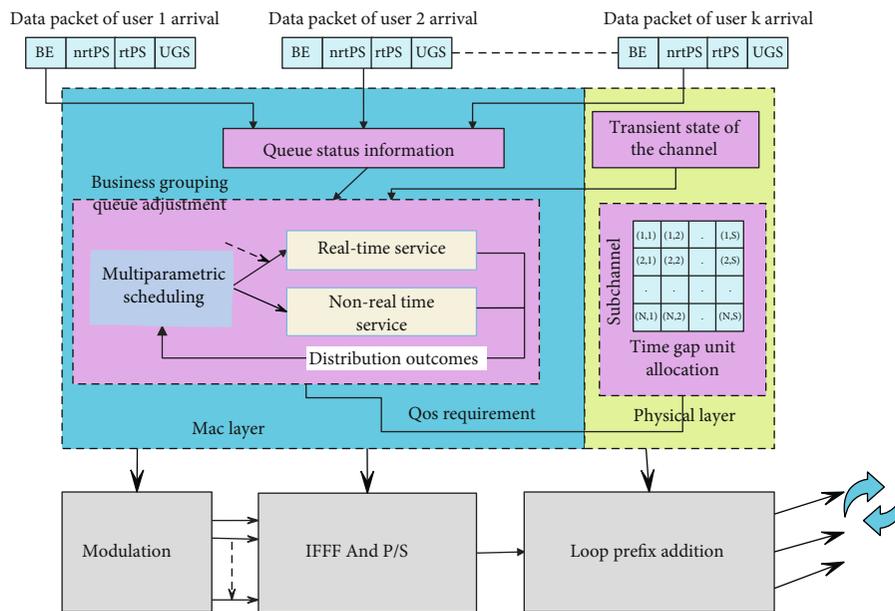


FIGURE 7: Multiservice system scheduling model.

on the cross-layer multi-ervice scheduling model diagram shown in Figure 7, the scheduling algorithm divides users into emergency users and nonemergency users according to different parameters of the MAC layer and the physical layer. Then, it adaptively sorts the packets according to the relationship between the user’s information and the maximum packet delay that can be tolerated by the user’s service. The groups are divided into emergency user compensation group, emergency user noncompensation group, nonemer-

gency user compensation group, and nonemergency user noncompensation group. The physical layer sorts the carrier resources and then allocates subchannels and corresponding time slots to each packet according to the sorting results provided by the scheduling algorithm, the channel state information of each user, the number of packets in the user buffer, and the QoS requirements of the service. Finally, the number of packets sent by each user is fed back to the scheduling model.

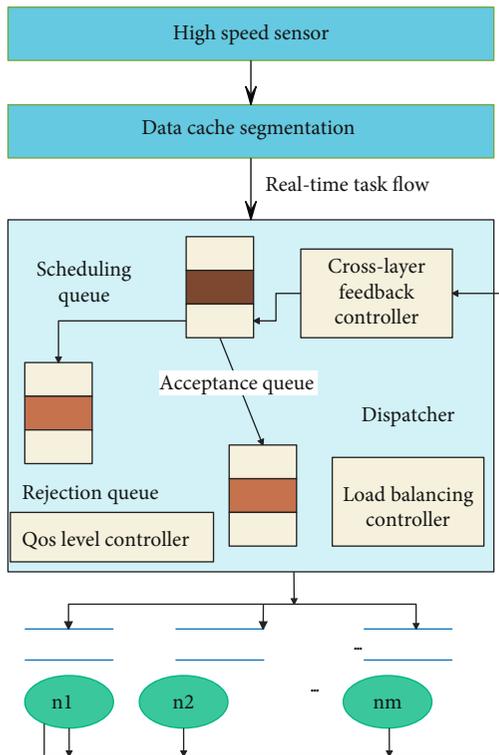


FIGURE 8: Multidimensional crosslayer scheduling model.

Under normal circumstances, high-speed sensor network nodes not only have the ability to sense and collect data but also have strong storage and processing capabilities. The real-time communication signal collected by the high-speed sensor node is first divided into multiple data blocks (tasks), and these tasks can be distributed to each high-speed sensor node for parallel processing. Nodes have certain storage and processing capabilities. Due to the real-time requirements of the task, the high-rate sensor network system not only needs the correctness of the task processing result but also ensures that the task can be completed within the specified time, and the final completion time of the task depends on the completion time of the last subtask. When the task is divided into multiple subtasks, all tasks of these complex applications need to be reasonably scheduled and allocated to the processing units of heterogeneous high-speed sensor nodes through a certain scheduling strategy, and task scheduling with the minimum completion time of the entire task is pursued. The quality of the scheduling algorithm directly affects the throughput rate of the task system, the success rate of scheduling, load balancing, and the QoS of the task. Scheduling algorithms for real-time tasks can be divided into two categories: static priority scheduling algorithm and dynamic priority scheduling algorithm.

The predetermined strategy of static priority scheduling is to schedule before the system starts running. Strict static scheduling cannot reschedule tasks while the system is running. The static scheduling algorithm has the advantages of

simple implementation, low scheduling overhead, and good predictability when the system is overloaded. Generally, a static scheduling algorithm is used to schedule periodic tasks. The dynamic priority scheduling algorithm dynamically allocates the priority of tasks, and its purpose is to have greater flexibility in resource allocation and scheduling. The dynamic scheduling algorithm is usually used to schedule aperiodic (aperiodic) tasks. Because the emergency hospitals used by high-rate sensor networks have a high degree of uncertainty, the time of task arrival, the size of the data, and other parameters are uncertain, and the data is burst; so, dynamic scheduling algorithms should be adopted. Scheduling algorithms for real-time tasks are divided into preemptive and nonpreemptive scheduling, that is, whether the task will be interrupted by other tasks during execution. In this paper, a nonpreemptive dynamic scheduling algorithm is used to meet the requirements, which can greatly reduce the switching cost between execution tasks, and is especially suitable for emergency hospitals with large data volume and strict delay requirements.

In a heterogeneous high-speed sensor network system, the same task can be allocated to different nodes for processing, different nodes have different processing capabilities, and the same task has different processing times on different nodes. The scheduling model can be divided into a distributed scheduling model and a centralized scheduling model. In the distributed scheduling model, the arrival of local tasks is independent and can be scheduled in parallel with others. In the centralized scheduling model, all tasks are centrally processed in the scheduling processing unit. In this paper, a centralized scheduling model is adopted. Compared with distributed scheduling, the centralized scheduling model has two notable features. (1) The centralized scheduling model is simpler and easier than the distributed scheduling model. (2) By using backup scheduling, it is convenient to realize fault-tolerant scheduling. In order to achieve the scheduling goal, this paper proposes a scheduling model based on crosslayer information on the basis of the traditional centralized scheduling model, as shown in Figure 8.

The schematic diagram of the communication between the sensor network and the IPv6 host is shown in Figure 9. IP-based wireless sensor networks are easier to achieve data interaction than other heterogeneous networks that use specific protocols. The Internet gateway enables end-to-end communication between sensing nodes and Internet hosts.

Figure 10 shows the network topology structure diagram generated by the BA model in a circular monitoring area with a radius of 500 m, and 200 nodes are randomly deployed. The initial network consists of three interconnected nodes. At each subsequent time step, a new node and two links are added to the network. In Figure 10, circles represent nodes, black nodes are selected backbone network nodes, and dotted lines represent communication links between nodes.

On the basis of the above model, the effect of the IoT sensor parameter regression processing is performed on

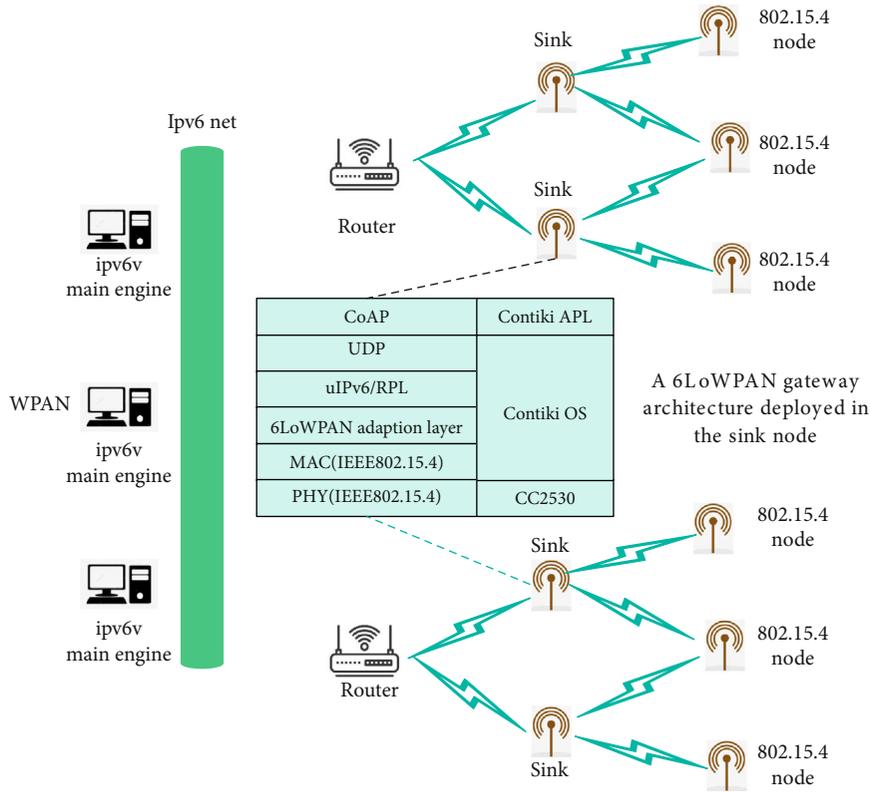


FIGURE 9: Schematic diagram of the communication between the sensing node and the host.

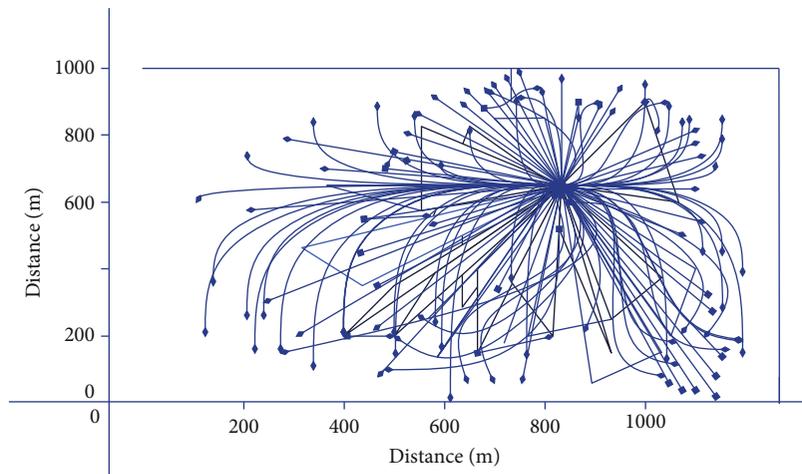


FIGURE 10: Topological structure diagram.

the model in this paper, and the results shown in Table 1 are obtained.

From the effect of IoT sensor parameter regression processing, the effect evaluation of the IoT sensor parameter regression method proposed in this paper is above 85; so, it has a good effect.

From the above research, it can be seen that the non-parametric regression model proposed in this paper can play a certain role in the sensor parameter regression of the Inter-

net of Things. The load balancing control effect of this model is verified on this infrastructure, and the results shown in Table 2 below are obtained.

From the load balancing effect, the evaluation of the load balancing control algorithm based on the nonparametric regression model of the Internet of Things link proposed in this paper is above 70; so, it has a good effect.

It can be seen from the above research that the IoT link load balancing control algorithm based on the

TABLE 1: Parameter regression processing effect of IoT sensors.

No.	Regression processing	No.	Regression processing	No.	Regression processing
1	89.43	18	91.64	35	92.95
2	88.94	19	90.41	36	94.91
3	90.57	20	87.55	37	94.25
4	88.25	21	94.54	38	85.99
5	91.28	22	87.49	39	90.23
6	86.36	23	86.15	40	89.30
7	91.22	24	85.81	41	95.88
8	88.62	25	85.19	42	85.78
9	92.72	26	89.15	43	85.16
10	92.48	27	94.11	44	88.09
11	91.90	28	92.19	45	92.55
12	89.48	29	90.50	46	95.43
13	89.28	30	89.68	47	85.64
14	87.38	31	87.41	48	87.28
15	85.65	32	85.24	49	86.58
16	95.01	33	92.62	50	94.45
17	93.56	34	94.32	51	90.50

TABLE 2: Evaluation of load balancing effect.

No.	Load balancing	No.	Load balancing	No.	Load balancing
1	71.82	18	82.38	35	79.52
2	68.98	19	88.59	36	79.19
3	88.17	20	83.01	37	82.58
4	89.62	21	86.47	38	83.30
5	88.69	22	89.10	39	66.47
6	71.73	23	88.69	40	66.30
7	83.45	24	77.16	41	83.59
8	79.97	25	89.37	42	80.13
9	78.78	26	85.60	43	75.10
10	88.82	27	67.69	44	68.21
11	85.77	28	80.33	45	85.59
12	80.09	29	70.69	46	73.88
13	85.27	30	72.45	47	78.26
14	68.70	31	68.75	48	69.68
15	76.42	32	88.52	49	66.93
16	81.77	33	77.57	50	78.96
17	69.17	34	75.55	51	75.55

nonparametric regression model proposed in this paper can effectively improve the internal scheduling of the IoT network link system and promote the load balancing effect.

5. Conclusion

The high-rate sensor network handles multiple mixed services including data, voice, video, web browsing, file transfer, and multimedia applications. In different applications, data, voice, and video information have different requirements for

transmission quality parameters such as bandwidth and delay. The amount of data in high-rate sensor networks is relatively large, various types of services have different requirements for the network QoS mechanism, and the resource constraints of sensor nodes are relatively more severe. Therefore, ensuring the QoS requirements of multiple services while meeting the energy constraints of high-rate sensor networks has put forward an urgent need for the optimization of the overall performance of high-rate sensor networks. How to coordinate the characteristic parameters of each sublayer in the high-rate sensor network to improve the overall network performance is a new challenge to the hierarchical protocol design of traditional wireless sensor networks. The effect evaluation of the IoT sensor parameter regression method proposed in this paper is above 85, the evaluation of the load balancing control algorithm based on the nonparametric regression model of the Internet of Things link proposed in this paper is above 70, and this paper combines the nonparametric regression model to improve the Internet of Things link load balancing algorithm, builds an intelligent model, and verifies the effectiveness of this model through research.

Data Availability

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

Conflicts of Interest

The author declares no competing interests.

Acknowledgments

This study is sponsored by the Henan University of Animal Husbandry and Economy.

References

- [1] J. Pan, Q. Xie, H. Chiang et al., "From the nature for the nature": an eco-friendly antifouling coating consisting of poly (lactic acid)-based polyurethane and natural antifoulant," *ACS Sustainable Chemistry & Engineering*, vol. 8, no. 3, pp. 1671–1678, 2020.
- [2] B. Behroozpour, P. A. M. Sandborn, M. C. Wu, and B. E. Boser, "Lidar system architectures and circuits," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 135–142, 2017.
- [3] J. Barowski, M. Zimmermanns, and I. Rolfes, "Millimeter-wave characterization of dielectric materials using calibrated FMCW transceivers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 8, pp. 3683–3689, 2018.
- [4] Y. Jiang, S. Karpf, and B. Jalali, "Time-stretch LiDAR as a spectrally scanned time-of-flight ranging camera," *Nature Photonics*, vol. 14, no. 1, pp. 14–18, 2020.
- [5] L. J. Xu, X. Lin, Q. He, M. Worku, and B. Ma, "Highly efficient eco-friendly X-ray scintillators based on an organic manganese halide," *Nature Communications*, vol. 11, no. 1, pp. 1–7, 2020.
- [6] Q. Y. Cheng, X. L. Zhao, Y. X. Weng, Y. D. Li, and J. B. Zeng, "Fully sustainable, nanoparticle-free, fluorine-free, and robust

- superhydrophobic cotton fabric fabricated via an eco-friendly method for efficient oil/water separation,” *ACS Sustainable Chemistry & Engineering*, vol. 7, no. 18, pp. 15696–15705, 2019.
- [7] N. Maring, P. Farrera, K. Kutluer, M. Mazzera, G. Heinze, and H. de Riedmatten, “Photonic quantum state transfer between a cold atomic gas and a crystal,” *Nature*, vol. 551, no. 7681, pp. 485–488, 2017.
- [8] H. Mohapatra and A. K. Rath, “Detection and avoidance of water loss through municipality taps in India by using smart taps and ICT,” *IET Wireless Sensor Systems*, vol. 9, no. 6, pp. 447–457, 2019.
- [9] A. Seri, G. Corrielli, D. Lago-Rivera et al., “Laser-written integrated platform for quantum storage of heralded single photons,” *Optica*, vol. 5, no. 8, pp. 934–941, 2018.
- [10] Z. Meng, J. Li, C. Yin et al., “Dual-band dechirping LFM CW radar receiver with high image rejection using microwave photonic I/Q mixer,” *Optics Express*, vol. 25, no. 18, pp. 22055–22065, 2017.
- [11] A. Al-Halafi and B. Shihada, “UHD video transmission over bidirectional underwater wireless optical communication,” *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–14, 2018.
- [12] H. C. Kumawat and A. B. Raj, “Extraction of Doppler signature of micro-to-macro rotations/motions using continuous wave radar-assisted measurement system,” *IET Science, Measurement & Technology*, vol. 14, no. 7, pp. 772–785, 2020.
- [13] Y. Wang, L. Ma, Y. Xu, and W. Xiang, “Computationally efficient energy optimization for cloud radio access networks with CSI uncertainty,” *IEEE Transactions on Communications*, vol. 65, no. 12, pp. 5499–5513, 2017.
- [14] F. Zhang, Q. Guo, and S. Pan, “Photonics-based real-time ultra-high-range-resolution radar with broadband signal generation and processing,” *Scientific Reports*, vol. 7, no. 1, pp. 1–8, 2017.
- [15] T. Zhong and P. Goldner, “Emerging rare-earth doped material platforms for quantum nanophotonics,” *Nano*, vol. 8, no. 11, pp. 2003–2015, 2019.
- [16] Z. Sabouri, A. Akbari, H. A. Hosseini, A. Hashemzadeh, and M. Darroudi, “Eco-friendly biosynthesis of nickel oxide nanoparticles mediated by okra plant extract and investigation of their photocatalytic, magnetic, cytotoxicity, and antibacterial properties,” *Journal of Cluster Science*, vol. 30, no. 6, pp. 1425–1434, 2019.
- [17] T. Zhong, J. M. Kindem, J. G. Bartholomew et al., “Nanophotonic rare-earth quantum memory with optically controlled retrieval,” *Science*, vol. 357, no. 6358, pp. 1392–1395, 2017.
- [18] K. Soga and L. Luo, “Distributed fiber optics sensors for civil engineering infrastructure sensing,” *Journal of Structural Integrity and Maintenance*, vol. 3, no. 1, pp. 1–21, 2018.
- [19] Y. J. Ma, A. Tadros, J. Du, and E. Y. Chang, “Quantitative two-dimensional ultrashort echo time magnetization transfer (2D UTE-MT) imaging of cortical bone,” *Magnetic Resonance in Medicine*, vol. 79, no. 4, pp. 1941–1949, 2018.
- [20] S. Sharaf and M. E. El-Naggar, “Eco-friendly technology for preparation, characterization and promotion of honey bee propolis extract loaded cellulose acetate nanofibers in medical domains,” *Cellulose*, vol. 25, no. 9, pp. 5195–5204, 2018.
- [21] C. Zhang, P. Xiao, F. Ni et al., “Converting pomelo peel into eco-friendly and low-consumption photothermic biomass sponge toward multifunctional solar-to-heat conversion,” *ACS Sustainable Chemistry & Engineering*, vol. 8, no. 13, pp. 5328–5337, 2020.