

Research Article

An IPv6 Passive-Aware Network Routing Algorithm Based on Utility Value Combined with Deep Neural Network

Zhiping Wan ¹, Zhiming Xu ¹, Jiajun Zou ¹, Shaojiang Liu ¹, Weichuan Ni ²,
and Shitong Ye ³

¹School of Information Science, Guangzhou Xinhua University, Dongguan 523133, China

²Facility Management & Laboratory Division, Guangzhou Xinhua University, Dongguan 523133, China

³Department of Data Science, Guangzhou Huashang College, Guangzhou 511300, China

Correspondence should be addressed to Shitong Ye; yst888_0@126.com

Received 23 October 2021; Accepted 12 November 2021; Published 28 November 2021

Academic Editor: Gengxin Sun

Copyright © 2021 Zhiping Wan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Passive sensing networks can maintain the operation of the network by capturing energy from the environment, thereby solving the energy limitation problem of network nodes. Therefore, passive sensing networks are widely used in data collection in complex environments. However, the complexity of the network deployment environment makes passive sensing nodes unable to obtain stable energy from the surroundings. Therefore, better routing strategies are needed to save network energy consumption. In response to this problem, this paper proposes an IPv6 passive-aware network routing algorithm for the Internet of Things. This method is based on the characteristics of passive sensing networks. By analyzing the successful transmission rate of the network node transmission link, transmission energy consumption, end-to-end transmission delay, and waiting delay of IPv6 packets, the utility evaluation function of the route is obtained. After the utility evaluation function is obtained, the network routing is selected through the utility evaluation function. Then, the utility value and the deep neural network method are combined to train the classification model. The classification model assigns the best routing strategy according to the characteristics of the current network, thereby improving the energy consumption and delay performance of the network.

1. Introduction

The Internet of Things is the expanding application and network extension of the communication network and Internet. It uses perception technology and intelligent devices to perceive and identify the physical world. It is interconnected by means of network transmission to complete the functions of numerical operation, signal processing, and information mining, so as to realize the information interaction and seamless connection between people, things and things, and between people and things, so as to achieve the purposes of real-time control, accurate management, and scientific decision-making of the physical world [1, 2]. The application range of the Internet of Things in the real world is very wide, including application fields such as smart home, vehicle network, underwater detection, human health monitor-

ing, and industrial monitoring [3, 4]. The current research on the Internet of Things mainly revolves around the active perception network; that is, the nodes of the Internet of Things are equipped with power supplies by default, and there is no need to consider the energy problem of the Internet of Things nodes. However, the active perception network is no longer sufficient to meet people's actual application needs. Passive energy supply is an important link to truly solve the space and time constraints of the Internet of Things application and realize the large-scale application of the Internet of Things.

The passive sensing network refers to a network composed of passive sensing nodes. Its nodes are not equipped with themselves or are not mainly dependent on their own power supply equipment for power supply, but support their computing, sensing, communication, and networking by

obtaining energy from the environment [5, 6]. Since passive nodes can maintain the operation of the network by capturing the energy of the surrounding environment, they can adapt to many application scenarios with limited energy supply and are a very promising network form in the Internet of Things. However, the passive sensing network can capture the energy of the surrounding environment, but it does not mean that it can obtain energy supply stably for a long time. For example, for passive sensing nodes that rely on optical energy, they will also lack energy in the face of weak optical signals. Therefore, when selecting network routes, we still need to pay attention to the transmission energy loss of the network [7]. Moreover, passive sensing nodes are often deployed in complex environments to perform monitoring tasks, so the network transmission delay will be relatively high under the influence of environment and terrain. Therefore, the routing propagation delay of passive sensing networks is also a problem that needs to be paid attention to when studying routing protocols.

In the research of the passive network of the Internet of Things, Hadi et al. proposed a general QoS-aware scheduling program for passive optical networks. In this research, the author discussed the service differentiation dynamic bandwidth allocation scheme in time division and wavelength division multiplexing passive optical. In the network application, in order to further reduce the computational complexity, the optimized closed-form solution involved in each scheduling iteration is derived and the transmission delay is directly included in the scheduling, which effectively reduces the transmission delay [8]. Li-ting et al. proposed a passive network architecture of an optical data center with high throughput and low delay, using passive optical devices such as an arrayed waveguide grating router, coupler, and demultiplexer, and gave wavelength allocation and packet transmission methods for each scale of architecture, with lower delay and higher throughput [9]. Hong-Chao et al. propose an opportunistic routing protocol with energy consumption and delay balance in passive sensing networks. The protocol estimates the expected energy consumption of the node by analyzing the node communication process, so that the node selects the neighbor node with low energy consumption as the forwarding candidate. The protocol makes decisions by combining the duty cycle information of the next hop neighbor node of the candidate node, so that the transmitting node can select the candidate node that can forward data faster to reduce the delay, so as to achieve the balance of energy consumption and delay performance [10].

This paper mainly focuses on the routing transmission energy consumption and transmission delay of passive sensing nodes in the Internet of Things. In the second section, the successful transmission rate, transmission energy consumption, transmission delay, and waiting delay of IPv6 packets are analyzed, and the utility evaluation function of routing is obtained. In the third section, a model is trained by using the method of the deep neural network and the method of utility value, so that it can select the best routing strategy for the current network. The fourth section carries out simulation experiments on the research methods of this paper and analyzes the results.

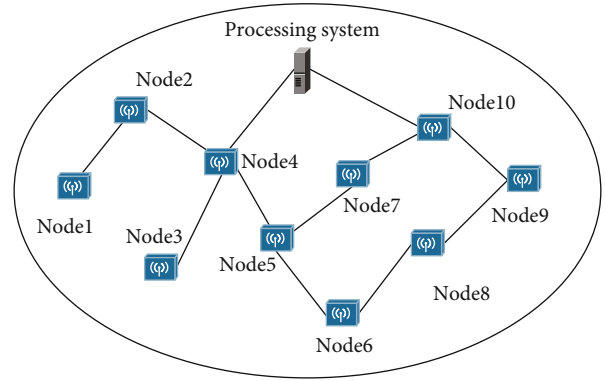


FIGURE 1: 10-node passive awareness network example.

2. Node Communication Link Analysis

In the passive sensing network of the Internet of Things, IoT devices located in the IPv6 network often need to send control/query commands to specific nodes in the passive sensing network, and these commands are encapsulated in IPv6 data packets. For example, the IoT center that performs monitoring tasks can send IPv6 packets to passive-aware network nodes deployed in complex environments through the IPv6 network to control the corresponding nodes to perform specific operations. This type of application requires that the transmission delay and transmission energy consumption be reduced as much as possible on the premise of reliable delivery of IPv6 packets. Figure 1 is a schematic diagram of a 10-node passive sensing network. The IoT node communicates with neighboring nodes within its communication radius to form a communication link, and there is a processing system in the network as the IoT center through the communication link sends control/query commands to IoT nodes.

2.1. Node Link Successful Transmission Rate. Using $(node_i, node_j)$ to represent the communication link between the i -th Internet of Things node $node_i$ and the j -th Internet of Things node $node_j$ in the network, the bit error rate of link $(node_i, node_j)$ is

$$\begin{aligned} BRE_{i,j} &= \frac{1}{2\pi} \int_a^{\infty} e^{-t^2/2} dt, \\ a &= \sqrt{\frac{2(P_t - P_{loss})B_N}{R}}, \end{aligned} \quad (1)$$

where $P_{loss} = P_l + 10\gamma \log_{10}(d_{i,j}) + P_t$.

P_l represents the power loss per unit distance, P_t represents the received power threshold of the device, and the received power threshold of all devices is the same by default. $d_{i,j}$ represents the distance between the i -th IoT node $node_i$ and the j -th IoT node $node_j$, and γ represents the power attenuation coefficient.

In this network, assuming that IPv6 packets are sent using a dynamic allocation strategy, if $A_{i,j}$ is used to

represent the number of fragments used by node $node_i$ in link $(node_i, node_j)$, the probability of success that node $node_i$ successfully transmits a data packet to $node_j$ by link $(node_i, node_j)$ for

$$p_{ij} = (1 - BRE_{i,j})^{K_{v6}/A_{i,j}}, \quad (2)$$

where K_{v6} represents the length of the IPv6 packet.

2.2. Transmission Energy Consumption. For passive sensing networks, IoT nodes maintain the operation of the network by capturing energy from the environment. For example, nodes can perceive and capture energy from the surrounding environment such as sunlight, temperature, wind, and RF signals, so as to support the operation of IoT devices and solve the energy limitation problem of IoT. However, limited by the comprehensive impact of a complex environment, nodes cannot stably obtain energy from the surroundings for a long time. Therefore, reducing transmission loss and saving energy as much as possible is an important standard to evaluate node routing performance for passive sensing networks.

For the energy loss in the transmission link $(node_i, node_j)$, we use E_{ij} to represent the unit transmission loss of the link, which is expressed as

$$E_{ij} = \alpha^{-1}e_0 + \beta^{-1}e_1D_i^2, \quad (3)$$

where α represents the transmission loss coefficient, e_0 represents the energy consumption of the node sending or receiving 1 bit on the circuit, e_1 represents the amplifier energy consumption for sending 1 bit of data under the communication radius D_i of the node i , and β represents the amplifier loss coefficient.

Then, a coded packet of length s is transmitted, and the sending energy consumption of node $node_i$ is

$$E_{ij}^t(s) = sE_{ij}. \quad (4)$$

The receiving energy consumption of node $node_j$ is

$$E_{ij}^r(s) = s\alpha^{-1}e_0. \quad (5)$$

Considering that the IPv6 packet adopts a dynamic allocation strategy, it needs to be decoded when $A_{i,j}$ coded packets are correctly received, so that node $node_j$ can successfully receive a complete IPv6 packet. Therefore, here, we consider the transmission energy consumption in the case of successfully receiving an IPv6 packet. If the success probability of node $node_i$ successfully transmitting a data packet to node $node_j$ through the $(node_i, node_j)$ link is p_{ij} , the total transmission energy consumption of an IPv6 packet from node $node_i$ sending to node $node_j$ successfully receiving is

$$*E_{ij}^t = \frac{A_{i,j}(E_{ij} + \alpha^{-1}e_0)}{p_{ij}}. \quad (6)$$

2.3. End-to-End Transmission Delay. For passive sensing networks used in emergency scenarios such as disaster monitoring or battlefield environment monitoring systems, the requirements for link transmission delay are very high. Therefore, in order to construct a better node route, we also need to consider the transmission delay problem.

The transmission delay for node $node_i$ which transmits a data packet of length $l_{i,j}$ to node $node_j$ is

$$\text{delay}_{ij} = t_p + \frac{l_{i,j}}{t_v}(1 + \varepsilon d_{i,j}), \quad (7)$$

where the subscript of $l_{i,j}$ means transfer from node $node_i$ to node $node_j$, t_p represents the time required for a node to compete for channels and encode data before sending data packets, t_v represents the transmission rate of unit data, $l_{i,j}/t_v$ represents the transmission duration, and ε represents the loss factor per unit propagation distance.

We consider that $A_{i,j}$ coded packets need to be correctly received before they can be decoded and reassembled into a complete IPv6 packet. According to the successful transmission probability p_{ij} of the data packet of the $(node_i, node_j)$ link, the delay from sending an IPv6 packet from node $node_i$ to node $node_j$ successfully receiving is

$$* \text{delay}_{ij} = \frac{t_p + (A_{i,j}/t_v)(1 + \varepsilon d_{i,j})}{p_{ij}}. \quad (8)$$

2.4. Time Delay of Waiting IPv6 Packets. Because passive nodes may not be able to transmit IPv6 packets successfully due to insufficient energy, they need to wait until sufficient energy is captured from the surrounding environment to continue to complete the transmission of IPv6 packets. Therefore, for the transmission process of data packets, we also need to consider the waiting delay when the remaining energy of nodes is insufficient.

We assume that the remaining energy of node $node_i$ is E_i^p and that node $node_i$ needs to successfully send k IPv6 packets to node $node_j$. If the remaining energy is insufficient, then the amount of energy that $node_i$ needs to capture is $E_{ij}^p(k)$:

$$E_{ij}^p(k) = k \frac{A_{i,j}E_{ij}}{p_{ij}} - E_i^p. \quad (9)$$

The electric energy that a node can obtain from the surrounding environment during the duration t is

$$E_c(t) = \delta \eta t, \quad (10)$$

where δ represents the charging efficiency coefficient of the node capacitor and η represents the average energy capture rate of the node.

According to formulas (9) and (10), when the remaining energy of node $node_i$ is E_i^p and k IPv6 packets need to be successfully sent to node $node_j$, the required waiting delay t_{ij} is

$$t_{ij} = \frac{k(A_{i,j}E_{ij}/p_{ij}) - E_i^p}{\delta\eta}. \quad (11)$$

2.5. Routing Utility Evaluation Function. In order to enable the selected node routing strategy to comprehensively consider the performance of the link successful transmission rate, transmission energy consumption, transmission delay, and waiting delay of the passive sensing network of the Internet of Things, we adopt an optimal routing evaluation function to determine the utility value of node routing; a routing strategy with a higher utility value is more suitable for the current network. Taking the (node_{*i*}, node_{*j*}) link as an example, we express the routing utility evaluation function as

$$Q_{ij} = \frac{w_1 p_{ij}}{w_2 * E_{ij} + w_3 * \text{delay}_{ij} + w_4 t_{ij}}. \quad (12)$$

$w_1, w_2, w_3,$ and w_4 are the weighting factors of the link successful transmission rate, transmission energy consumption, transmission delay, and waiting delay, respectively, which can be specifically set according to the performance requirements of the network. E_{ij} represents the total transmission energy consumption of the IPv6 packet from the sending of node_{*i*} to the successful reception of node_{*j*}, D represents the delay of the IPv6 packet from the sending of node_{*i*} to the successful reception of node_{*j*}, and the transmission delay calculated by t_{ij} includes the required when the remaining energy of the node is insufficient charging time.

Assuming that the route is (node₁, node₂, ..., node_{*n*}), the best route evaluation function of the route is expressed as

$$\overline{Q}_{1n} = \frac{\sum_{i=1, j=i+1}^{n-1} Q_{ij}}{n-1}. \quad (13)$$

Through the best route evaluation function, we can determine the utility value \overline{Q}_{1n} of the route (node₁, node₂, ..., node_{*n*}). We can choose the node routing strategy according to the utility value. The route with the higher utility value is more likely to be selected as the best routing strategy.

However, currently, there are many routing strategies for the Internet of Things commonly used. To select the best routing strategy for the current network through the utility value method, the amount of calculation involved is very large. Therefore, in the following chapters, we use the deep neural network method, using specific network instances as input, 150 routing strategy IDs as labels, and the routing strategy ID with the largest utility value as the true label of the instance, to perform the deep neural network training. After the neural network model is trained, when a new network instance is inputted, the optimal routing strategy can be selected without performing utility value calculation.

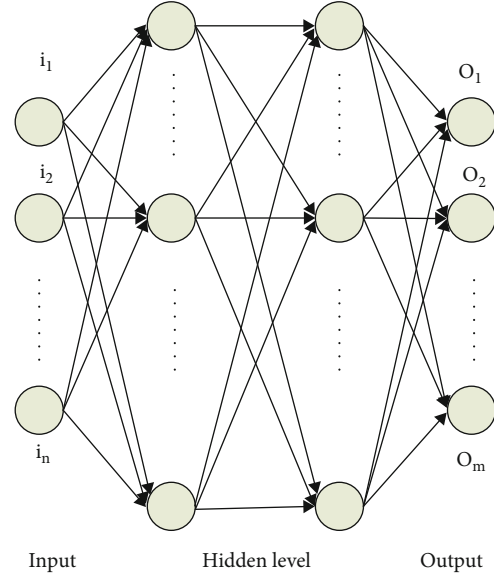


FIGURE 2: Deep neural network structure.

3. Routing Selection Based on Deep Neural Network

A deep neural network (DNN) is a neural network structure composed of a large number of neurons through an input layer, an output layer, and multiple hidden layers (usually at least two hidden layers). DNN has achieved great success in common tasks such as natural language processing, image processing, and other major machine learning problems [11, 12]. At present, the research field of the computer network also uses DNN technology to optimize the network. In the research of this paper, our node routing is selected by a deep neural network, and the neural network is trained by introducing a feedforward deep neural network and back propagation learning algorithm [13, 14]. In a feedforward network, information flows from the input node to the output node through the network without any feedback/loop connection. In the back propagation, the network model is optimized through the gradient optimization algorithm. The structure of the deep neural network is shown in Figure 2.

3.1. Neural Network Structure Used. In this algorithm, we construct the input layer node by taking the number of network nodes, node communication radius, network state, successful transmission probability of link, transmission energy consumption, end-to-end transmission delay of path, and waiting delay of the IPv6 packet as the characteristics of the instance. Taking the ID number of the optimal routing strategy corresponding to the example as the real output, the routing strategy is predicted by constructing the feedforward propagation from the input layer to the hidden layer and from the hidden layer to the output layer, and the parameters are learned by back propagation.

In order to make the constructed deep neural network play a role in routing selection, we used the characteristics

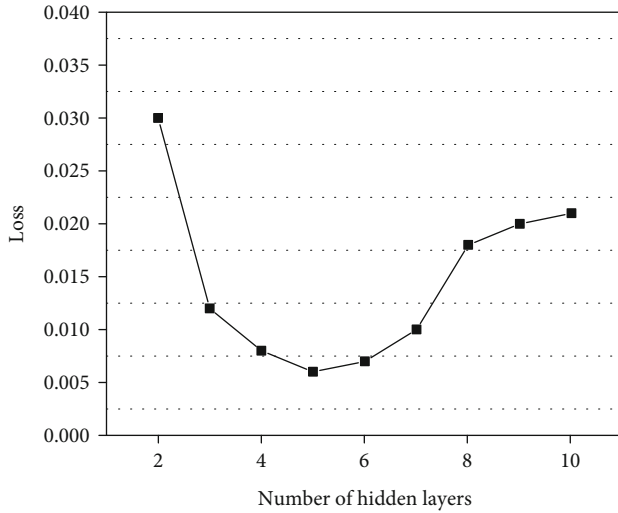


FIGURE 3: Model loss under different hidden layers.

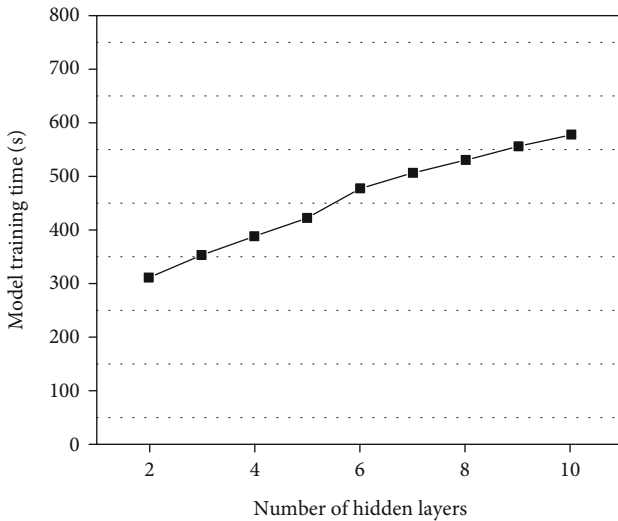


FIGURE 4: Model training time under different hidden layers.

of the data set, and the structure configuration of the constructed deep neural network model is as follows:

- (1) *Network input layer*: the nodes of the input layer are determined according to the feature number of the data set. The input layer nodes are connected with the first layer hidden layer nodes, and the activation function of Relu is used. The form of the Relu function is as follows:

$$f(x) = \max(0, \infty). \quad (14)$$

x indicates that the Relu function gets inputted.

- (2) *Network output layer*: the output layer uses the softmax function. The softmax function is often used in the multiclass structure of deep neural networks to normalize all output results. Each instance can only

belong to one class (that is, a certain routing strategy):

$$S_i = \frac{e_i}{\sum_{i=1}^m e_i}, \quad (15)$$

where e_i represents the i -th output result; there are m outputs in total.

- (3) *Network hidden layer*: we use 8 hidden layers. The number of nodes in the hidden layer can be adjusted according to the number of input characteristics of network instances. All hidden layers use Relu activation function.
- (4) *Network loss function*: the loss function we use is "cross-entropy loss." Cross-entropy loss is very effective for estimating the loss of multiple classification methods. The form of the cross-entropy loss function used is as follows:

$$H(X) = - \sum_{i=1}^n p(x_i) \log(q(x_i)), \quad (16)$$

where x_i represents a specific instance and $p(x_i)$ represents the real label. In this article, when the label of the instance belongs to the real category, $p(x_i) = 1$. When the label of the instance does not belong to the true category, $p(x_i) = 0$. $q(x_i)$ represents the prediction result after the instance is processed by the model.

- (5) *Network optimization method*: in this deep neural network model, we use Adam as the network optimizer. The learning rate is the default learning rate of the "Adam Optimizer," which is 0.001.
- (6) *Number of network iterations*: the network model has been trained for multiple iterations in this article.
- (7) *Index*: the performance evaluation index used is accuracy. The accuracy rate represents the proportion of the number of correctly classified instances to the total number of instances.
- (8) *Verification data*: in order to verify the performance of the model, a verification split of 0.2 is used in this article; that is, we use 20% of the training data to verify the network model.

The flow of the entire algorithm is as follows:

- (1) Select multiple routing strategies to build a routing set C
- (2) For the data set composed of network instances, the label of each network instance is a routing policy in C , and the real label is the routing policy with the largest utility value of the network instance in C , wherein the utility value of the network instance is obtained according to the routing utility evaluation function of formula (12)

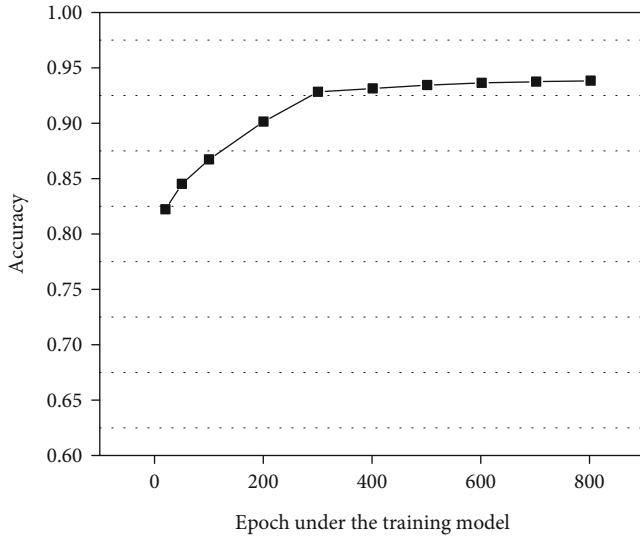


FIGURE 5: Model accuracy under different iterative training times.

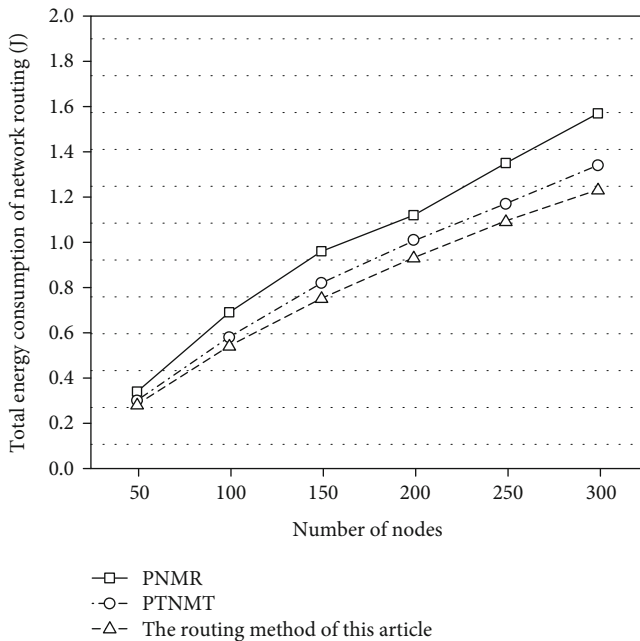


FIGURE 6: Total energy consumption of network routing under different numbers of nodes.

- (3) After obtaining the real label of each network instance in the data set, the data set is divided into a training set, verification set, and test set, and is sent to the deep neural network model for training. The trained model can select the routing strategy with the maximum utility value for the network instance from the routing set C under the condition of a given network instance

3.2. Data Set Scheme. For the data set used for deep neural network model training, we use the OMNET++ simulator to obtain the simulated data set, which contains 50,000 instance samples. Each network instance sample contains

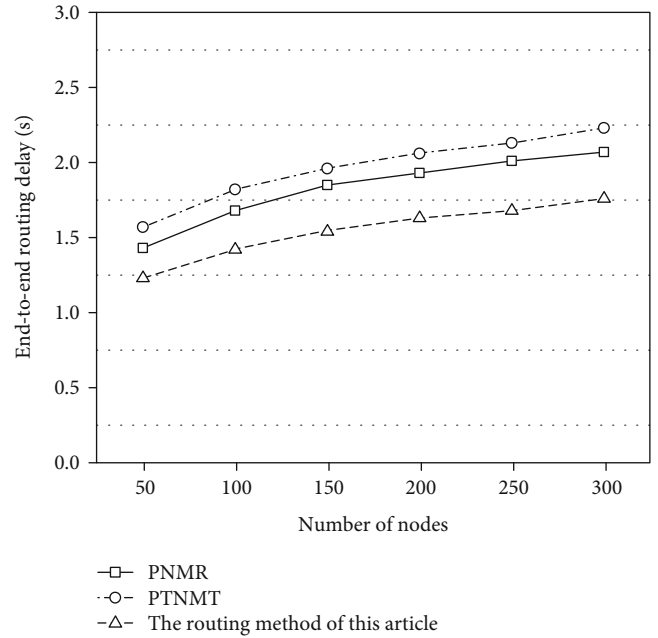


FIGURE 7: Total end-to-end routing delay under a different number of nodes.

several characteristics of the number of network nodes, node communication radius, network status, link's successful transmission probability, transmission energy consumption, end-to-end transmission delay of the path, and waiting delay of IPv6 packets. We equip the entire data set with 150 routing strategies as the classification result; that is, each network instance corresponds to an optimal routing strategy as the true label. The best routing strategy is determined according to the best routing evaluation mechanism proposed in this paper, which can make the routing strategy with the greatest utility value the best routing strategy. The 150 routing strategies include routing with minimum energy consumption, routing with minimum delay, general IoT node routing algorithms, and heuristic algorithms. The model obtains the best weights through training, so that when we enter a new network instance, we can base on the number of network nodes, node communication radius, network status, link's successful transmission probability, transmission energy consumption, and path. The end-to-end transmission delay and the waiting delay of IPv6 packets output an optimal routing strategy.

4. Experimental Simulation Results

Before getting the final usable deep neural network model, we first analyze the influence of different hidden layers on the model in the experimental test link, so as to determine an optimal number of hidden layers. We test the loss and training time of the model according to the number of hidden layers, and test the accuracy of our model under different iteration times. We show the results of the model on the test data set. The deep neural network model adopts the PyTorch framework and is implemented using Python language programming. The PC configuration used in the

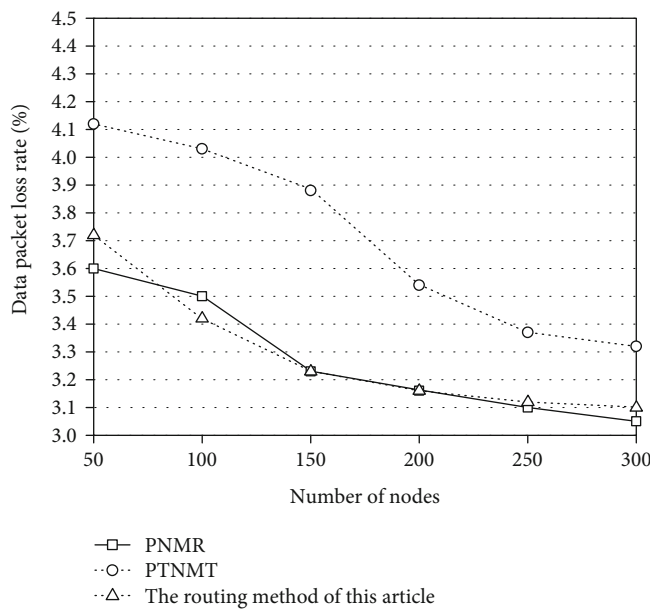


FIGURE 8: Average data packet loss rate.

experiment includes NVIDIA GeForce RTX 3080 Ti 12 GB GDDR6 video memory, Intel i9 processor, 32 G memory. Different hidden layers are all experimented on the same computer. The result obtained is shown in the following figure.

Figure 3 shows how the training loss of the model changes with the number of hidden layers. It can be seen from the figure that the training loss gradually increases after 5 hidden layers. In deep neural networks, the number of hidden layers is not as large as possible. Sometimes, using more hidden layers than required by the model will cause the model's classification ability to decrease. Therefore, for this article, using a 5-layer hidden layer deep neural network to achieve routing strategy selection will get better results.

It can be observed from Figure 4 that the training time of the model will increase as the number of hidden layers increases. Since we set the same number of neurons in each hidden layer, as the number of hidden layers increases, the number of neurons will also increase. In a deep neural network, each neuron has a weight, so more neurons will increase the amount of weight calculation, so the training time will increase as the number of hidden layers increases.

Figure 5 shows the accuracy of the model on the test set after training under different iterations of the training set. It can be seen from the figure that when the training set is iterated to 300 times, the accuracy of the model converges to a certain value, and then, the accuracy value will not change significantly.

Through the above experiments, we have determined that the deep neural network model of this article uses 5 hidden layers and only iterates 300 times during the training process to optimize the model parameters to obtain our final model results. In order to verify the effectiveness of the trained model in node routing, we use the trained model and other algorithms to compare the total energy consumption of routing transmission and the total end-to-end trans-

mission delay. The comparison algorithms are the passive network multipath routing proposed in the literature [15] (here, we abbreviate it as PNMR for the convenience of expression) and the passive label network multihop routing protocol proposed in the literature [16] (here for the convenience of expression, we referred to it as PTNMT) for comparison. Among them, literature [15] conducts network multipath routing detection from link average delay and load balancing, and literature [16] considers the problems of asymmetry of communication links and transmission interference in passive sensing networks.

We first conducted a comparison experiment on routing energy consumption. With different numbers of nodes, we let the system randomly select the source node and the destination node, and then, we let the three comparison algorithms choose the node route by themselves, so that the source node can successfully transmit the IPv6 packet to the destination node. Figure 6 shows the total energy consumption of the PNMR method, PTNMT method, and the method in this paper under a different number of nodes in the network routing. It can be seen from the figure that as the number of network nodes increases, the total energy consumption of the network routing of the three algorithms continues to increase. When the number of network nodes is small, the total energy consumption of the routing of the three algorithms is relatively close. When the number of network nodes increases, the total energy consumption of routing in this paper will be less, because the deep neural network model of this paper will select the best routing strategy for the current network to reduce end-to-end transmission loss.

In another group of comparative experiments, we also let the system randomly select the source node and destination node under different number of nodes, then let the three comparison algorithms select the node route by themselves, and record the delay time when they successfully transmit

IPv6 packets from the source node to the destination node. As can be seen from the results in Figure 7, for Figure 7, the end-to-end routing delay of the network may increase with the increase of the number of nodes. This is because when the number of nodes increases, and the source node and destination node are randomly selected by the system, the number of routing hops from the source node to the destination node may need to be more and the delay will be greater. In Figure 7, we can see that the method in this paper has less end-to-end routing delay than the other two algorithms.

The following figure shows the comparison of the average data packet loss rate of the three algorithms. It can be seen from the results in Figure 8 that as the number of nodes increases, the distribution of the average data packet loss rate of the network will gradually decrease and tend to be flat. Since the source node and the destination node are randomly selected by the system, when the number of nodes is small, the probability of link interruption may increase, resulting in a higher network packet loss rate. From the comparison of the three algorithms, it can be seen that the data packet loss rate of the algorithm in this paper is close to that of the PNMR algorithm, and the packet loss rate of the PTNMT algorithm is smaller.

5. Conclusions

In the Internet of Things, a passive sensing network can capture energy from the surroundings through Internet of Things devices, so as to solve the problem of energy limitation when the Internet of Things is deployed in the field environment. However, the uncertainty of the field environment makes the Internet of Things devices unable to capture stable energy, and node routing still needs to reduce energy loss as much as possible. Therefore, this paper studies an IPv6 passive sensing routing strategy selection method for the Internet of Things, which is aimed at reducing the data transmission energy consumption and transmission delay of node routing, and at improving the operation efficiency of the network. Combined with the method of the deep neural network, this paper intelligently selects routing strategies for the current network through an artificial intelligence model. Simulation results show that the proposed method can reduce the transmission energy consumption and transmission delay of the network.

Data Availability

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

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the Characteristic Innovation Projects of Ordinary Universities in Guangdong Province (2018KTSCX319) and Key Discipline Project Guangzhou Xinhua University (No. 2020XZD02).

References

- [1] F. Piccialli, S. Cuomo, N. Bessis, and Y. Yoshimura, "Data science for the Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4342–4346, 2020.
- [2] D. Cao, L. Li, C. Marina, L. Chen, Y. Xing, and W. Zhuang, "Special issue on Internet of Things for connected automated driving," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3678–3680, 2020.
- [3] F. Li, K. Y. Lam, X. Li, Z. Sheng, J. Hua, and L. Wang, "Advances and emerging challenges in cognitive Internet-of-Things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5489–5496, 2020.
- [4] Z. Wan, Z. Xu, S. Liu, W. Ni, and S. Ye, "An Internet of Things roaming authentication protocol based on heterogeneous fusion mechanism," *IEEE Access*, vol. 8, pp. 17663–17672, 2020.
- [5] Z. Zhou, "The connection engine of the Internet of Things: passive sensing network," *Computer Science and Application*, vol. 10, no. 11, pp. 2089–2094, 2020.
- [6] S. Denis, B. Bellekens, A. Kaya, R. Berkvens, and M. Weyn, "Large-scale crowd analysis through the use of passive radio sensing networks," *Sensors (Basel, Switzerland)*, vol. 20, no. 9, pp. 2624–2632, 2020.
- [7] H. Hwang, J. H. Lim, J. H. Yun, and B. Jeong, "Pattern-based decoding for Wi-Fi backscatter communication of passive sensors," *Sensors*, vol. 19, no. 5, pp. 1157–1176, 2019.
- [8] M. Hadi, C. Bhar, and E. Agrell, "General QoS-aware scheduling procedure for passive optical networks," *Journal of Optical Communications and Networking*, vol. 12, no. 7, pp. 217–226, 2020.
- [9] X. Li-ting, Z. Shao-biao, and J. Xu-han, "Research of two optical data center network frameworks with high throughput and low latency," *Study on Optical Communications*, vol. 222, no. 6, pp. 21–24, 2020.
- [10] H.-C. Gao, X.-J. Chen, D. Xu, Y. Peng, Z.-Y. Tang, and D. Y. Fang, "Balance of energy and delay opportunistic routing protocol for passive sensing network," *Journal of Software*, vol. 30, no. 8, pp. 2528–2544, 2019.
- [11] G. E. Dahl, Dong Yu, Li Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio Speech & Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
- [12] S. Liu, K. Hu, W. Ni, Z. Xu, F. Wang, and Z. Wan, "A cognitive relay network throughput optimization algorithm based on deep reinforcement learning," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 2731485, 8 pages, 2019.
- [13] X. Zhang, Y. He, and S. Jian, "Channel pruning for accelerating very deep neural networks," in *IEEE International Conference on Computer Vision*, IEEE Computer Society, 2017.
- [14] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *International Conference on Computer Aided Verification*, Springer, Cham, 2017.

- [15] S. Pan, Y. Wang, M. Xiong, L. Gu, and G. Hu, "Detection of multipath routing with passive delay measurements," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, 2019.
- [16] C. Liu, Z. J. Haas, and Z. Tian, "On the design of multi-hop tag-to-tag routing protocol for large-scale networks of passive tags," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1035–1055, 2020.