Chen, C., Li, J., Balasubramaniam, V., Wu, Y., Zhang, Y., & Wan, S. (2021). Contention Resolution in Wi-Fi 6-Enabled Internet of Things Based on Deep Learning. IEEE Internet of Things Journal, 8(7), 5309–5320.

https://doi.org/10.1109/JIOT.2020.3037774

Contention Resolution in Wi-Fi 6-Enabled Internet of Things Based on Deep Learning

Chen Chen, *Senior Member, IEEE*, Junchao Li, Venki Balasubramaniam, *Member, IEEE*, Yongqiang Wu, Yuru Zhang, and Shaohua Wan, *Senior Member, IEEE*

***Abstract*—Internet of Things (IoT) is expected to vastly increase the number of connected devices. As a result, a multitude of IoT devices transmit various information through wireless communication technology, such as the Wi-Fi technology, cellular mobile communication technology, low-power wide-area network (LPWAN) technology. However, even the latest Wi-Fi technology is still ready to accommodate these large amounts of data. Accurately setting the contention window (CW) value significantly affects the efficiency of the Wi-Fi network. Unfortunately, the standard collision resolution used by IEEE 802.11ax networks is nonscalable; thus, it cannot maintain stable throughput for an increasing number of stations, even when Wi-Fi 6 has been designed to improve performance in dense scenarios. To this end, we propose a CW control strategy for Wi-Fi 6 systems. This strategy leverages deep learning to search for optimal configuration of CW under different network conditions. Our deep neural network is trained by data generated from a Wi-Fi 6 simulation system with some varying key parameters, e.g., the number of nodes, short interframe space (SIFS), distributed interframe space (DIFS), and data transmission rate. Numerical results demonstrated that our deep learning scheme could always find the optimal CW adjustment multiple by adaptively perceiving the channel competition status. The finalized performance of our model has been significantly improved in terms of system throughput, average transmission delay, and packet retransmission rate. This makes Wi-Fi 6 better adapted to the access of a large number of IoT devices.**

***Index Terms*—Contention window (CW) optimization, deep learning, Internet of Things (IoT), Wi-Fi 6.**

I. Introduction

INTERNET of Things (IoT) is currently a booming research and application area, which aims to connect everyday physical things to a global-scale network via wireless data communication and identification technologies, such as sensors and radio-frequency identification (RFID). It revolutionizes a large number of data-driven applications in a wide variety of real-world domains. IoT-based solutions potentially provide competitive advantages in various key sectors of society, e.g., smart cities, smart homes/smart building management, healthcare, security, transportation, and surveillance [1]–[4]. In general, one of the key parts of the IoT design focuses on the wireless communication system that acts as a data-bridge to offload the computation, storage, and access requirements from a large number of diversified applications and control messages. However, although evolving at the 5G era, the existing cellular technologies still lack the ability to support a large amount of data exchange from many battery-powered devices spread over a wide area [5], [6]. Even with the ongoing narrow-band IoT (NB-IoT) technology, which claims to support a deployment density of 200 000 NB-IoT devices within a cell, the network performance is still not satisfied after several field tests due to the overwhelming computations, complex signal fading, and differentiated Quality-of-Service (QoS) requirements, especially diversified data or service models. As a result, the standardization of the IoT network is still an open issue and a hot topic even when many technologies are already popular in specific application scenarios, such as LoRa, eMTC, NB-IoT, Internet of Vehicles (IoVs), and Wi-Fi [7]–[11].

On the contrary, as a mature technology with more than 17 years of development and standardization, Wi-Fi had remarkable success in the deployment of IoT networks in indoor and outdoor environments during the last decade. In addition, Wi-Fi is currently not only the most common Internet access technology but has also been expanded across a wide variety of markets, including consumer, mobile, and automotive. Although Wi-Fi has achieved great success in business, research, and daily usage, the existing Wi-Fi standard is still unable to provide satisfactory network access performance for numerous IoT devices with diversified service requirements. Fortunately, the IEEE organization recently started a new task group, i.e., the IEEE 802.11ax or Wi-Fi 6, to investigate and deliver next-generation Wi-Fi technologies for the scenarios of dense IoT networks with a large number of data traffics. Different from the previous Wi-Fi standard, the Wi-Fi 6 standard uses popular technologies,

Manuscript received September 12, 2020; revised October 20, 2020; accepted November 6, 2020. Date of publication November 16, 2020; date of current version March 24, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1807500; in part by the National Natural Science Foundation of China under Grant 62072360, Grant 62001357, Grant 61672131, and Grant 61901367; in part by the Key Research and Development Plan of Shaanxi Province under Grant 2017ZDCXL-GY-05-01 and Grant 2020JQ-844; in part by the Key Laboratory of Embedded System and Service Computing (Tongji University) under Grant ESSCKF2019-05; in part by the Ministry of Education, Xi'an Science and Technology Plan under Grant 20RGZN0005; and in part by the Xi'an Key Laboratory of Mobile Edge Computing and Security under Grant 201805052-ZD3CG36. *(Corresponding author: Shaohua Wan.)*

Chen Chen, Junchao Li, and Yuru Zhang are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: cc2000@mail.xidian.edu.cn; 519435930@qq.com; zyr15848264587@163.com).

Venki Balasubramaniam is with the School of Science, Engineering and Information Technology, Federation University, Mount Helen, VIC 3350, Australia (e-mail: v.balasubramanian@federation.edu.au).

Yongqiang Wu is with the Department of Management, Zhejiang Wellsun Intelligent Technology Company Ltd., Hangzhou 317200, China (e-mail: wyq@wellsun.com).

Shaohua Wan is with the School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China (e-mail: shaohua.wan@ieee.org).

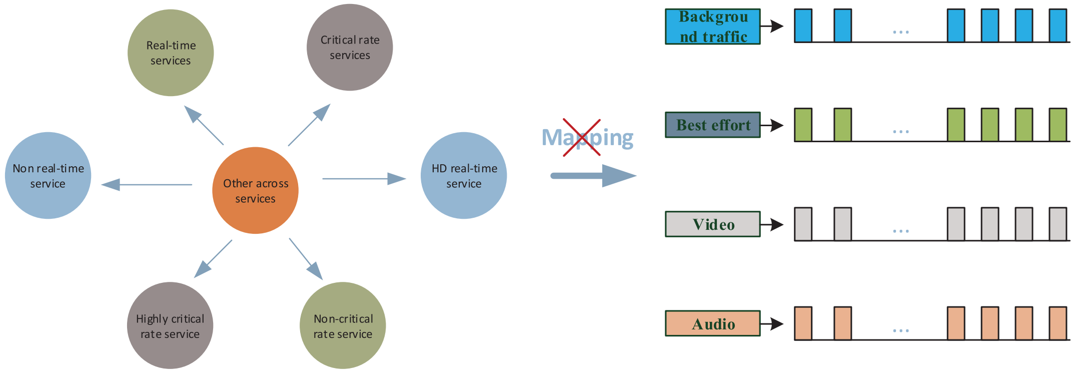

Fig. 1. Multiservice versus four ACs.

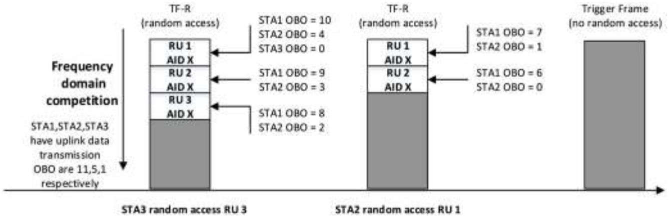

Fig. 2. OFDMA UL random access.

such as multiuser, multiple-input, multiple-output technology (MU-MIMO), orthogonal frequency-division multiple access (OFDMA), dedicated channel allocation, and machine learning to increase the physical bit rate [12]. At the same time, Wi-Fi 6 achieves efficient multiuser access by reducing the frame error rate (FER) and improving the spectrum reuse rate to provide a better user experience [13]–[15].

Although Wi-Fi 6 is expected to increase the actual throughput by four times in a dense user environment,the requirement for backward compatibility with the original carrier sense multiple access with collision avoidance (CSMA/CA) mechanism, this makes the MAC layer incompetent in the multiuser OFDMA PHY in downlink (DL) and uplink (UL) cases. In addition, as depicted in Fig. 1, the original access category (AC) configuration based on coarse service classification, i.e., best-effort, background traffic, video, and voice cannot satisfy the diversification of services especially in the IoT paradigm. As a result, a fine-grained contention resolution is necessary to match the service diversification in IoT networks.

In fact, in Wi-Fi 6, when the STA receives the trigger frame for random access (TF-R frame), it uses the UL OFDMA backoff mechanism (UL-OFDMA Backoff, OBO) to compete for the resource unit (RU), as shown in Fig. 2. Similar to the original CSMA/CA scheme, a backoff counter is randomly generated in the OFDMA contention window (OCW) before attempting to send packets, and the channel can only be accessed when the backoff counter decreases to 0.

Fig. 2 shows an example of UL random access [16]. The figure shows that STA1, STA2, and STA3 have UL data to send, and their OBO counters are 11, 5, and 1, respectively. In the first attempt of random access, the available resources are RU1–RU3. After the backoff process, STA3's OBO counter is initially decreased to 0, and the RU3 is randomly assigned to STA3. In the second round of random access, STA2's OBO is initially reduced to 0, and the RU1 is randomly selected for access. This example indicates that the backoff mechanism for 802.11ax has no difference with the common approach, where the contention window (CW) obeys the binary exponential backoff (BEB) policy. The BEB simply exponentially doubles the OCW value at collisions to avoid repeated collisions. However, it always resets the OCW value to obtain the minimum OCW ($OCW_{\min}$) after a successful transmission on the basis of a blind assumption of low-level network congestion. As a result, the 802.11ax high efficiency wireless local area networks (HEW) also suffers from an issue, where the traditional BEB scheme is unaware of the channel contention level and cannot provide optimized throughput in highly dense network environments. For a dense network, resetting the CW to its minimum size may result in more collisions and poor network performance. Similarly, for a small network environment, a blind increase in CW size may cause an unnecessarily long delay in accessing the channel.

On the contrary, although BEB is prone to severe data collisions in the dense environment, an intelligent backoff strategy can address the issue effectively [17]. Actually, many research works have proposed modifications to the BEB-based CSMA/CA, especially in IEEE 802.11ax networks. In [18], a self-scrutinized channel observation-based scaled backoff (COSB) mechanism is presented to handle the high-density contention challenges in 802.11ax networks. In [19], an adaptively scaled backoff (ASB) mechanism to mitigate the performance degradations of BEB in highly dense IEEE 802.11ax networks was proposed. In [20], a priority-based backoff scheme was envisioned to overcome the unexpected delay of time critical traffic, such as ultrahigh definition (UHD) video traffics. Although these previous efforts have emphasized the importance of an intelligent backoff scheme to the performance guarantee of 802.11ax networks, their fixed adjustment pattern or limited adaptivity when increasing and resetting the CW still induces performance degradation and cannot optimize the system output. To satisfy the diverse requirements of dense WLANs, prospective HEW is anticipated to autonomously access the best channel resources with the assistance of sophisticated wireless channel condition inference to control channel collisions [21]. Such intelligence is possible with the introduction of deep learning techniques into the Wi-Fi 6 systems.

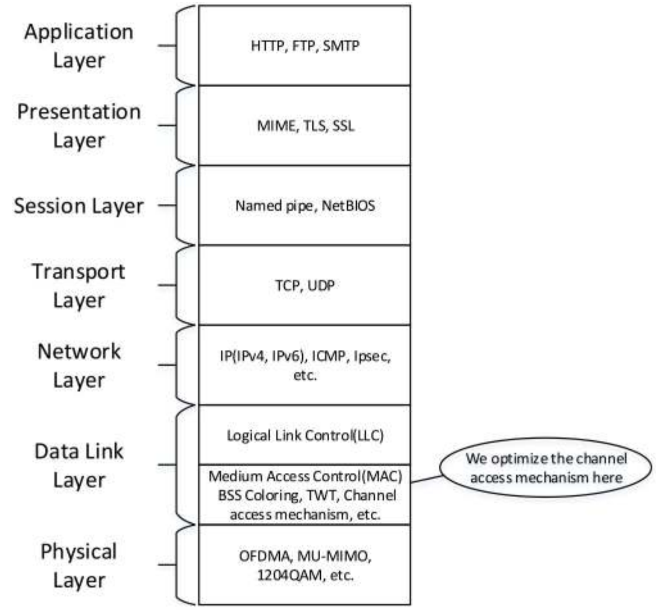
Application Layer
HTTP, FTP, SMTP
Presentation Layer
MIME, TLS, SSL
Session Layer
Named pipe, NetBIOS
Transport Layer
TCP, UDP
Network Layer
IP(IPv4, IPv6), ICMP, Ipsec, etc.
Data Link Layer
Logical Link Control(LLC)
Medium Access Control(MAC) BSS Coloring, TWT, Channel access mechanism, etc.
We optimize the channel access mechanism here
Physical Layer
OFDMA, MU-MIMO, 1204QAM, etc.

Fig. 3. IEEE 802.11ax each layer structure diagram.

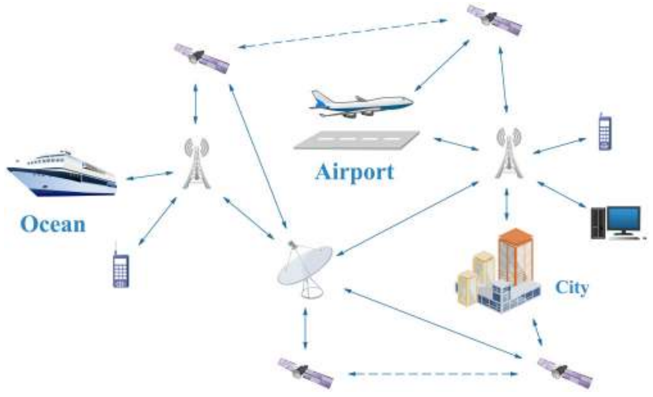

Fig. 4. Diagram of integrated air, space, ocean, and land communication.

Therefore, a BEB scheme based on deep learning is proposed to dynamically adjust the backoff strategy in accordance with the real-time system performance. As shown in Fig. 3, this study mainly optimizes the channel access mechanism of IEEE 802.11ax MAC. In summary, our model aims to optimize the system throughput, latency, and retransmission rate or identify the optimal tradeoff point among them by relying on the parameter optimization and feedback iterative capability of machine learning. The basic idea behind our work can be generalized as follows. Initially, we collected data from thousands of simulations in terms of system throughput, latency, and OCW multiple factors. Then, we conducted a loss function to evaluate the performance of our deep learning model on throughput, latency, and retransmission rate. Subsequently, we train our model weights with collected data and use the trained model to predict the system throughput, latency, and retransmission rate. An optimal backoff multiple factor is finally determined in accordance with the predicted performance indices. With our deep learning-based BEB scheme, the system performance of the 802.11ax network could be intentionally optimized in real-time and timely response to the channel variations as well as contention levels. Our work is to adaptively adjust the parameters of Wi-Fi 6, especially the parameters of access control to enable the traditional Wi-Fi 6 to adaptively work in more 6G future scenarios, as shown in Fig. 4. The figure shows the integrated communication scene of air, space, ocean, and land [22], [23].

Our contributions are summarized as follows.

1) A regression model between the OCW multiple factor and system performance indices is implemented in IoT systems by using the error backpropagation algorithm.
2) A deep learning model is established with the OCW multiple factor as the input and the system throughput, latency, and nodal retransmission rate as the output. Our model is then trained using the Bayesian regularization algorithm, and the mean square error (MSE) and curve fitting degree are considered to evaluate the model output. At the same time, we set the network convergence conditions, such as the lower limit of the MSE and the upper limit of the number of network iterations, to ensure that the training process is short.
3) The OCW multiple factor is predicted in accordance with the real-time network performance, by which the BEB policy of the 802.11ax network could be adaptively adjusted corresponding to the channel variations and packet contention levels.

Paper Framework: The remainder of our article is organized as follows. Section II reviews some relevant works on Wi-Fi MAC layer optimization with pros and cons given for IoT systems. Section III describes the procedure for training data acquisition and preprocessing. Our deep learning-based BEB model is presented in Section IV. Section V demonstrates the experimental configurations, numerical results, and corresponding performance analysis. Our article is concluded in Section VI.

II. Related Work

At present, studies on Wi-Fi MAC layer optimization are relatively mature. In the literature, research directions can be roughly divided into two categories, namely, traditional optimization methods and machine learning-based optimization methods.

A. Traditional Optimization Methods

Traditional optimization methods can be further divided into several categories, e.g., priority-based optimization algorithms, probability-based optimization algorithms, and adaptive optimization algorithms.

Priority-Based Optimization Algorithm: In [24], a MAC layer optimization algorithm for the TDMA/CSMA hybrid protocol is proposed, and its backoff window is calculated on the basis of the node priority and the number of node conflicts at the current moment. This algorithm divides the priority of nodes into four levels, with level 1 having the highest priority. This algorithm focuses on the network priority of the nodes. Thus, the nodes with higher priorities can have greater probability to occupy empty slots. However, the algorithm divides the priority into 1, 2, 3, and 4 levels, and the CW increase factor is simply increased in accordance with 1, 2, 3, and 4 times, without considering the actual transmission requirements.

Probability-Based Optimization Algorithm: In [25], a selection probability-based backoff algorithm (SPB) is proposed. The backoff mechanism becomes steeper when the network is idle, owing to the SPB algorithm, and the backoff mechanism is smoother when the network is crowded. This algorithm improves system fairness and optimizes system throughput. The increase and decrease in the backoff window of the SPB algorithm are controlled by the amplification index and reduction index. However, the amplification index and reduction index are fixed values, and they cannot be intelligently changed, following the network environment.

Adaptive Optimization Algorithms: In [26], a MAC layer optimization algorithm based on transmission limit adaptation is proposed. Its central idea is to ensure that nodes can complete the data interaction within three retransmission times. If the retransmission time is more than three, then the CW window value is increased by the linear backoff algorithm. In [27], the linear backoff algorithm is superior to the exponential backoff algorithm in terms of collision probability performance. This algorithm combines the linear backoff avoidance algorithm with the exponential algorithm, thereby reducing the system delay while reducing the probability of system conflicts and improving the fairness of the system. However, either in linear backoff or exponential backoff, the multiples of the increase and decrease in the backoff window of this algorithm are fixed values. Thus, the adaptability to different network environments is reduced. In [28], an optimization algorithm based on network load adaptively adjusting the backoff index is proposed. This algorithm uses a nonlinear curve fitting to predict the backoff index and assigns backoff index on the basis of network load and business priorities. It uses the smaller delay as the cost, improves the network throughput, reduces the packet loss rate, and optimizes the overall network performance. This algorithm uses curve fitting to predict the backoff index, but does not effectively measure the quality of the curve fitting, thereby inevitably reducing the reliability of the algorithm. In [29], an improved algorithm based on the hybrid-distributed coordination function (HBCDF) is proposed to represent the competition window. The HBDCF algorithm uses the transmission control protocol (TCP) congestion control mechanism, which can effectively reduce the DCF collision probability and network delay in larger networks, thereby improving the saturation throughput of the system. This simulation results show that the HBCDF algorithm cannot improve every standard of the network (such as collision probability, network delay, and throughput), thereby reducing its fairness. In addition, the algorithm has not yet reached a conclusion on the optimal threshold and optimal slope of different network scales, thereby limiting its performance improvement. In [30], a COSB mechanism for the CSMA/CA of high-efficiency WLANs is devised. The COSB guarantees high throughput and low delay by reducing the number of collisions during the channel access mechanism in saturated and unsaturated traffic environments. However, in a dense network environment, COSB suffers from the fairness problem because some STAs continuously operate at a higherCWsize, and few fortunate STAs can operate at a lower CW size. Under COSB, once the STA reaches a larger CW, it has to transmit successfully many times to return to the smaller CW. This approach seems diffcult in a dense network environment due to the high probability of collision.

B. Machine Learning-Based Optimization Methods

On the contrary, with the improvement of computing power and the promotion of the development of big data, algorithms based on machine learning and deep reinforcement learning are proposed to optimize MAC layer channel access.

Algorithms Based On Machine Learning: In [31], a framework called intelligent-CW (ICW) that allows nodes to adapt their $CW_{\min}$ values based on observed transmissions is proposed, ensuring they receive fair share of the channel airtime. The $CW_{\min}$ value at a node is set on the basis of a random forest, which is a machine learning model that includes a large number of decision trees. ICW achieves high throughput efficiency while maintaining fair allocation of the unlicensed channels with other neighboring nodes. One of the most important features of ICW is that its learning structures can be easily trained and optimized to ensure high spectrum efficiency. However, this algorithm does not consider the setting of the $CW_{\max}$ and CW increasing methods. In [32], a machine-learning-enabled EDCA (MEDCA) mechanism for QoS-supported MAC layer channel access in dense WLANs is proposed. This mechanism utilizes a Q-learning algorithm, which is one of the prevailing models of machine learning, to infer the network density and adjust its backoff CW accordingly. This MEDCA mechanism overcomes the limitations of EDCA by implementing a channel observation-based collision probability for the scaling of backoff parameters.

Algorithm Based On Deep Reinforcement Learning: In [33], an intelligent QL-based resource allocation (iQRA) mechanism is proposed for MAC layer channel access in dense WLANs. The proposed intelligent paradigm learns diverse WLAN environments and optimizes performance, compared with conventional nonintelligent MAC protocols. However, the performance of iQRA may degrade in small networks due to low and irregular rewards.

We have proposed a conflict avoidance optimization algorithm based on deep learning by analyzing the problems using these algorithms. This algorithm can effectively predict the system throughput, delay, and packet retransmission rate under different increase multiples of CW. Through the prediction results, we can select the appropriate CW increase factor to improve system performance.

III. Data Collection

This chapter simulates the IEEE802.11ax CSMA/CA network system. The original data of the experiment are obtained through MATLAB [34] simulation, including system throughput, system delay, and packet retransmission rate. Table I shows a brief description of the symbols used in the text. To ensure the accuracy and completeness of the data collection process, we reasonably set the simulation parameters of

TABLE I
LIST OF NOTATIONS

Notations	Description
S	System throughput
D_s	Amount of valid data successfully received by the terminal
T_k,T_z	Elapsed time when the transmission status is k or z
T	System delay
Ω,α	The state set
L	Current number of layers
m	Total number of training samples
s_{L-1}	The number of neurons in layer L-1
$w_{ij}^{(L)}$	The connection weight between the j-th neuron in the L-1 layer and the i-th neuron in the L layer
$h_j^{(L)}$	The output value of the jth neuron in the L layer
$b_i^{(L)}$	The offset term of the ith neuron in the Lth layer
$net_i^{(L)}$	Represents the transition term of the ith neuron in the Lth layer
$y_k(i)$	The predicted output value of the kth output neuron
$Y_k(i)$	The true value of the kth output neuron
$f()$	Activation function
E	Loss function for all training samples
$E(i)$	Loss function of a single training sample
η	Learning rate
R	Regression value
$\sum$	Summation symbol

TABLE II
SIMULATION PARAMETERS

Parameters	Explanation
Number of network nodes	20
Data packet number distribution	Poisson distribution
Time slot interval	20us
Simulation time	200s
SIFS	10us
DIFS	50us
Physical layer data transmission rate	2Mbit/s
Bit error rate	10^{-6}
CW_{min}	0.64ms
CW_{max}	20.48ms
Channel model	Nakagami-m decline[35]
Packet generation process	Poisson distribution[24]

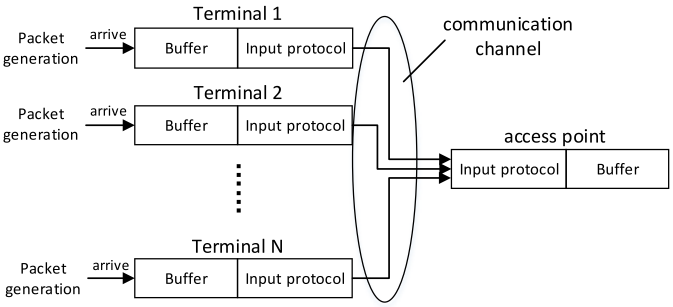

Fig. 5. Packet generation procedure.

the system. The final system simulation parameters are shown in Table II. The data packet generation process is shown in Fig. 5, where the input protocol is the CSMA/CA protocol.

To obtain the influence of the increase multiple of CW on the system output parameters, We use 0.1 times as the step size to expand the CW increase factor from 1.1 times to 10 times, resulting in 90 different CW increase factors. We simulate systems with different CW multiples and set the collection rate of system throughput, delay, and packet retransmission rate to 1s, that is, we collect the output data of the system every 1 s of time. Therefore, in the total simulation time of 200 s, 200 output data with time change will are obtained. Then, we repeat the experiment 1000 times to obtain 200 × 1000 data under one CW increase multiple. We average 200 × 1000 output data to obtain the final output data of system throughput, delay, and packet retransmission rate under different CW growth multiples. The definitions of system throughput, delay, and packet retransmission rate are as follows.

TABLE III
SAMPLED RAW DATA

CW enlargement multiple	Throughput (bits/s)	Delay (s)	Data packet retransmission rate (units/s)
1.1	868508.68	0.032359	305
1.2	975978.90	0.024821	303
1.3	1111732.67	0.017724	258
1.4	1232692.44	0.012971	202
1.5	1339715.15	0.009247	139

1) Throughput S is the amount of valid data successfully received by the terminal per unit time; it is expressed in

$$S = \frac{D_s}{\sum_{k \in \Omega} T_k} \tag{1}$$

where D_s is the effective amount of data successfully received by the terminal, T_k is the time considered when the transmission state is k, and Ω is the state set. The transmission status includes the following: waiting for transmission status, successful transmission status, and failed transmission status.

2) The delay T represents the time required for one data packet to travel from one end to the other, and it is expressed in formula

$$T = \sum_{z \in \alpha} T_z \tag{2}$$

where T_z is the time considered when the transmission state is z, and α is the state set. The transmission status here includes the following: queuing status, sending status, processing status, and propagation status.

3) The data packet retransmission rate is the number of data packets retransmitted by the terminal per unit time. The raw data example is shown in Table III.

The sample data in Table III show that the system throughput has a large order of magnitude. We know that when using MSE for error analysis, if the input parameter has a large order of magnitude, even if the final predicted value is very close to the true value, then the error result is maintained at a high order of magnitude. Thus, the magnitude of the error cannot be effectively reflected. At the same time, the larger magnitude of the input parameters also reduces the convergence speed of the network [36]. Therefore, prior to the deep learning, we need to normalize the data of throughput, delay, and packet retransmission rate to the range of [0,1] [37], and the examples of normalized processed data are shown in Tables IV–VI. This data set is unavailable for all application scenarios. The data set is collected in real time; thus, for different application scenarios, data are collected in real time.

Prior to the deep learning, we need to divide the data into different types. We divide the original data according to the

TABLE IV
EXAMPLE OF NORMALIZED THROUGHPUT

CW enlargement multiple	Throughput (bits/s)
1.1	0.306610695764182
1.2	0.437169780766391
1.3	0.602088843896627
1.4	0.749035569168779
1.5	0.879051001276918

TABLE V
EXAMPLE OF NORMALIZED AVERAGE TRANSMISSION DELAY

CW enlargement multiple	Delay (s)
1.1	1
1.2	0.752609123728257
1.3	0.519691499835904
1.4	0.363702001969150
1.5	0.241483426320971

TABLE VI
EXAMPLE OF NORMALIZED PACKET RETRANSMISSION RATE

CW	Packet retransmission rate (units/s)
1.1	1
1.2	0.993377483443709
1.3	0.844370860927152
1.4	0.658940397350993
1.5	0.450331125827815

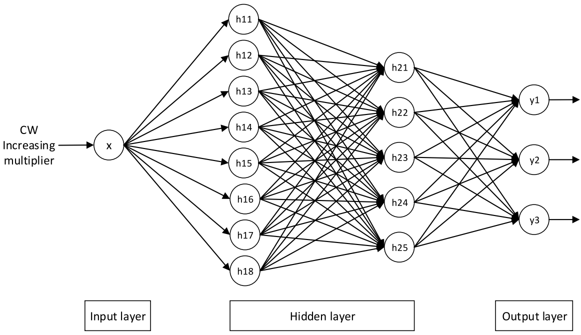

Fig. 6. Architecture of the proposed neural network.

ratio of 7:1.5:1.5, in which the training data account for 70% of the total data, and the verification data and the test data account for 15% of the total data, respectively. After the neural network model is established by deep learning, the output prediction data need to be denormalized to obtain the final prediction data value.

IV. PROPOSED MODEL

This section mainly introduces the proposed deep learning model. In Section IV-A, the architecture model of the neural network is introduced, the information of the input layer, hidden layer, and output layer of the network architecture are described, and the process of determining the number of hidden layers is discussed. In Section IV-B, the algorithm flow of the deep learning neural network is described in detail. The neural network architecture and the neural network algorithm flow are shown in Figs. 6 and 7, respectively.

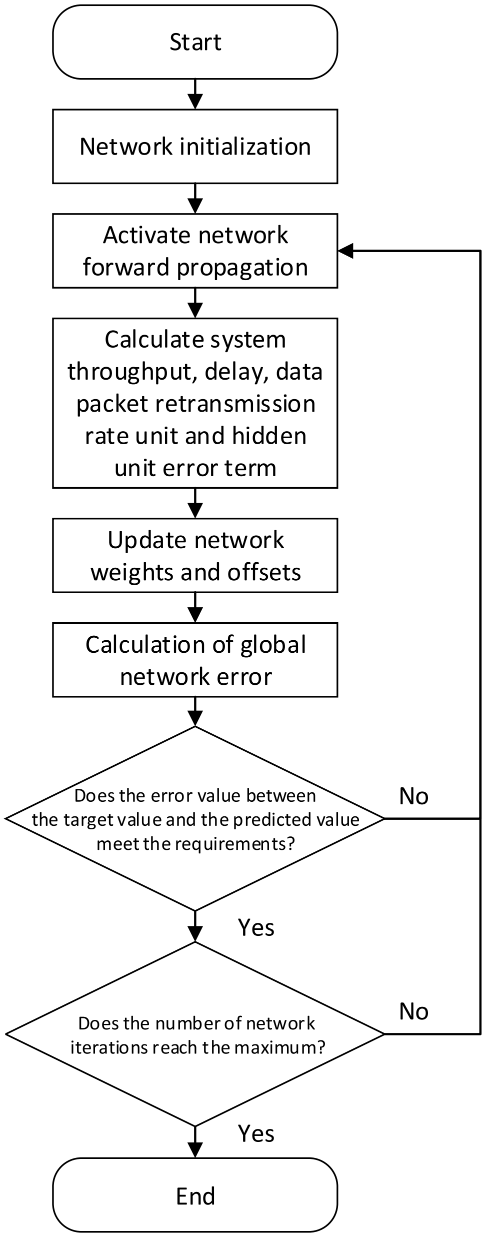
Start
Network initialization
Activate network forward propagation
Calculate system throughput, delay, data packet retransmission rate unit and hidden unit error term
Update network weights and offsets
Calculation of global network error
Does the error value between the target value and the predicted value meet the requirements?
No
Yes
Does the number of network iterations reach the maximum?
No
Yes
End

Fig. 7. Flowchart of the proposed CW adaptive model.

A. Introduction to Network Architecture

The proposed deep neural network model is built on the basis of feedforward neural network [38], [39]. We have optimized the network structure through multiple experimental simulations and selected the structure with the best experimental effect as the final network structure. Fig. 6 shows that this deep neural network has three layers, including an input layer, a double hidden layer, and an output layer [40]. The relationship between the input and output of each layer in this neural network is expressed by

$$\mathrm{net}_i^{(L)} = \sum_{j=1}^{s_{L-1}} w_{ij}^{(L)} h_j^{(L-1)} + b_i^{(L)} \tag{3}$$

$$h_j^{(L)} = f\left(\mathrm{net}_i^{(L)}\right) \tag{4}$$

TABLE VII
Performance of Different Number of Hidden Layers

Number of hiden layers	MSE	Regression diagram	Regression fitting curve
1 layer	5.27e-4	Medium	Poor
2 layer	5.25e-5	Excellent	Excellent
3 layer	4.47e-5	Excellent	General
4 layer	5.31e-5	Excellent	General
5 layer	5.62e-5	Excellent	General

where s_{L-1} represents the number of neurons in layer L-1; $w_{ij}^{(L)}$ represents the connection weight between the jth neuron in the L-1 layer and the ith neuron in the L layer; $b_i^{(L)}$ represents the offset term of the ith neuron in the Lth layer; $h_j^{(L-1)}$ represents the output of the jth neuron in the L-1 layer, and can also represent the input of the jth neuron in the L layer; $\mathrm{net}_i^{(L)}$ represents the transition term of the ith neuron in the Lth layer; and $f()$ represents the neuron activation function.

Input Layer: The input layer is a single input and the input parameter is OCW increase multiple.

Hidden Layer: This network has two hidden layers, namely, hidden layer 1 and hidden layer 2. Generally, the more number of hidden layers indicates a stronger learning ability of the network, more complex the network, and higher possibility of the overfitting phenomenon. In a single input network, the number of the hidden layers should not be extremely large. The simulation result shows that the network performance of two hidden layers is the best; thus, selected number of hidden layers is two. Table VII shows the performance comparative results of different hidden layers. This neural network sets the number of neurons in hidden layers 1–8 and that in hidden layers 2–5.

Table VII shows that the MSE of the hidden layer neural network with one layer is higher, and the regression fitting curve is very poor. The MSE of the hidden layer with four and five layers increases, and the effect of the regression fitting curve is general, which shows that the number of hidden layers is large. Thus, we should select the neural network with the number of hidden layers greater than 1 and less than 4. Although the MSE of the hidden layer with three-layer neural network is the lowest, the regression fitting curve is not as good as that of the two layers, and the number of MSE of the two layers satisfies the requirements. In a comprehensive consideration, we select the double hidden layer neural network.

Output Layer: The number of neurons in the output layer is three, and the output parameters are system throughput, system delay, and data packet retransmission rate.

B. Introduction of Algorithm Flow

As shown in Fig. 7, the algorithm flow is mainly divided into six steps. The steps are as follows.

Step 1: The weights, bias terms, and learning rates are initialized in the neural network.

Step 2: Forward propagation is activated, the expected value of the loss function is obtained with respect to system throughput, delay, and packet retransmission rate. The loss function is represented by

$$E = \frac{1}{m}\sum_{i=1}^{m} E(i) \tag{5}$$

where m is the total number of training samples and $E(i)$ is the loss function of a single training sample and is expressed by

$$E(i) = \frac{1}{2}\sum_{k=1}^{n} (Y_k(i) - y_k(i))^2 \tag{6}$$

where the coefficient (1/2) is set to facilitate the derivation of the equation, and the value is not fixed; n represents the number of output neurons in the output layer, and the output neurons of this deep neural network are system throughput, delay, and packet retransmission rate. Thus, $n = 3$; $Y_k(i)$ is the true value of the kth output neuron, and $y_k(i)$ is the predicted output value of the kth output neuron. Therefore, the loss function of this deep neural network can be expressed by

$$E = \frac{1}{2m}\sum_{i=1}^{m}\sum_{k=1}^{3} (Y_k(i) - y_k(i))^2. \tag{7}$$

Step 3: According to the loss function, the output unit error term and hidden unit error term are calculated. The output unit error term is the gradient value of the loss function with respect to the output element, and the hidden unit error term is the gradient value of the loss function with respect to the hidden element. Each unit error term can be further divided into a connection weight error term and a bias error term. The connection weight error term and offset error term of the output layer of a single training sample are expressed by equations (8) and (9), respectively

$$\begin{aligned}\frac{\partial E(i)}{\partial w_{zj}^{(L)}} &= \frac{\partial}{\partial w_{zj}^{(L)}}\left(\frac{1}{2}\sum_{k=1}^{n}(Y_k(i) - y_k(i))^2\right)\\ &= -(Y_z(i) - y_z(i))f'(x)\big|_{x=\mathrm{net}_z^{(L)}} h_j^{(L-1)}\end{aligned} \tag{8}$$

where $w_{zj}^{(L)}$ represents the connection weight of the zth neuron in the output layer (layer L) and the jth neuron in layer L-1; and z represents the system throughput, delay, and packet retransmission rate of this deep neural network. $f'(x)$ represents the derivative of the activation function. The types of activation functions of each layer of this deep neural network are shown in Table VIII

$$\begin{aligned}\frac{\partial E(i)}{\partial b_{z}^{(L)}} &= \frac{\partial}{\partial b_{z}^{(L)}}\left(\frac{1}{2}\sum_{k=1}^{n}(Y_k(i) - y_k(i))^2\right)\\ &= -(Y_z(i) - y_z(i))f'(x)\big|_{x=\mathrm{net}_z^{(L)}}\end{aligned} \tag{9}$$

where $b_z^{(L)}$ represents the offset value of the zth neuron in the output layer. The output layer of this deep neural network is the third layer, i.e., $L = 3$. We set $\delta_z^{(3)} = -.(Y_z(i) - y_z(i))f'(x)\big|_{x=\mathrm{net}_z^{(3)}}$; therefore, the connection weight error term and offset term error of the output layer can be further expressed as

$$\frac{\partial E(i)}{\partial w_{zj}^{(3)}} = \delta_z^{(3)} h_j^{(3-1)} \tag{10}$$

$$\frac{\partial E(i)}{\partial b_z^{(3)}} = \delta_z^{(3)}. \tag{11}$$

The calculation process of the connection weight error term and the offset error term of each hidden layer is the same and is not described here.

Step 4: The connection weights and bias term of the neural network are updated. The updated formulas are as follows:

$$w_{ij}^{(L)\prime} = w_{\bar{y}}^{(L)} - \eta \frac{\partial E}{\partial w_{ij}^{(L)}} \tag{12}$$

$$b_i^{(L)\prime} = b_i^{(L)} - \eta \frac{\partial E}{\partial b_i^{(L)}} \tag{13}$$

where a and b represent the updated weight and offset values, respectively; η represents the learning rate, and $\eta \in (0, 1)$.

Step 5: The global error of the network is calculated.

Step 6: Whether the error value satisfies the requirements or whether the number of training iterations is used up is determined. If any of these conditions is satisfied, then the output parameters are the best parameters at present; if all the conditions are not satisfied, then repeat steps 2–6.

V. Experimental and Numerical Results

A. Experimental Performance Measurement Index

The measurement index of network model performance.

1) The R value of regression, it is used to measure the correlation between the predicted value and the target value, and when the R value is 1, the predicted value of this deep neural network is closest to the target value.
2) MSE, used to determine whether the output of this deep neural network is accurate. If MSE $< 10^{-3}$, then this deep neural network is initially considered desirable.
3) The regression fitting curve of the neural network to original data. The regression fitting curve is one of the most important indices to evaluate the neural network. The higher curve fit indicates more accurate data prediction result and more reliable network.

B. Experimental Scene, Parameter Table, and Network Topology

MATLAB is used to establish the proposed feedforward neural network architecture scenario, and the experimental simulation is conducted in accordance with the network topology introduced in Section IV. The detailed information on the network topology are presented in Section IV. The experimental parameters are shown in Table VIII, where logsig is a logarithm function of type S, tansig is a hyperbolic tangent function, trainbr is a Bayesian regularization algorithm, and learngd is a learning function of gradient descent weight.

TABLE VIII
Experimental Parameters

Parameter type	Parameter value
Max number of iterations	1000 times
MSE target value	10^{-6}
Initial learning rate	0.01
Momentum factor	0.9
Normalized capability test value	16
Activation function f1	logsig
Activation function f2	tansig
Activation function f3	purelin
Learning function	learngd

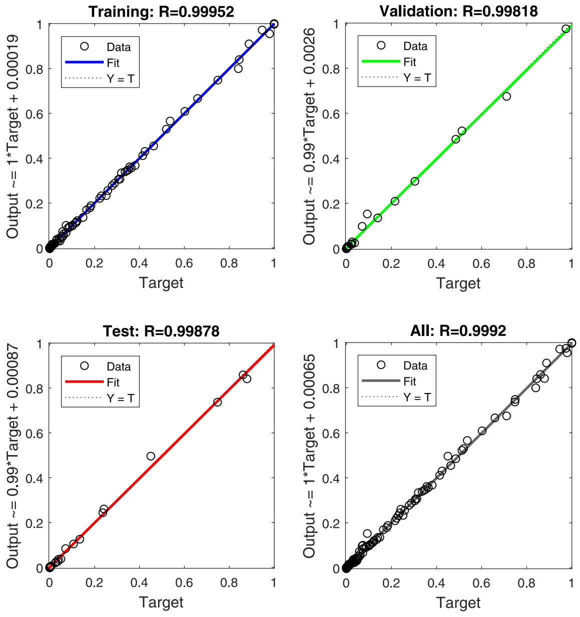
Training: R=0.99952
Validation: R=0.99818
Test: R=0.99878
All: R=0.9992

Fig. 8. Regression performance of the proposed neural network.

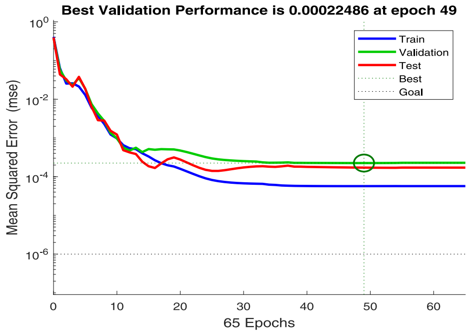
Best Validation Performance is 0.00022486 at epoch 49

Fig. 9. Training performance of the proposed neural network.

C. Experimental Simulation Results

D. Experimental Result Specification

Fig. 8 shows a network regression diagram. The figure shows that the R value of the verification data is 0.99818, and the R value of the test data is 0.99878. The correlation between the output data of the neural network and the target data has exceeded 99.8%, which indicates that this network can predict the output result accurately.

Fig. 9 shows that the MSE no longer decreases when the number of iterations of this neural network algorithm is 49. At this time, the MSE of the verification data is 2.2486e-4, which is already lower than 10e-3, and the MSE of the training data is less than 10e-4, thereby satisfying the requirement of MSE.

Fig. 10 shows the comparison between the original data of system throughput and the predicted data of neural network

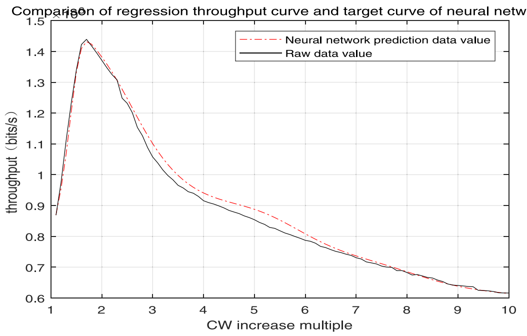

Fig. 10. System throughput with different CW.

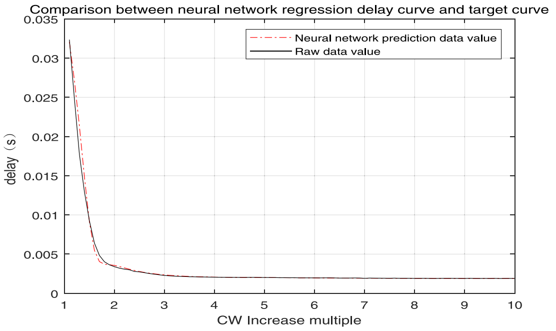

Fig. 11. Average transmission delay with different CW.

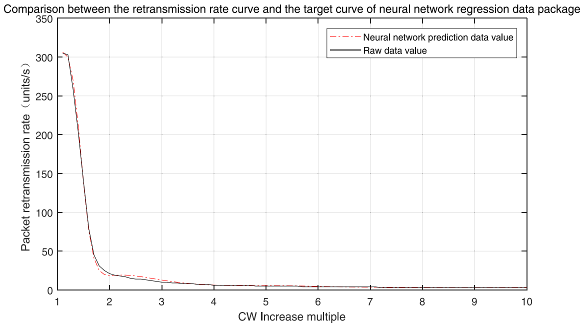

Fig. 12. Packet retransmission rate with different CW.

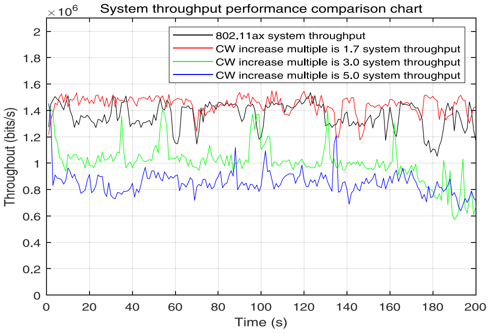

Fig. 13. System throughput.

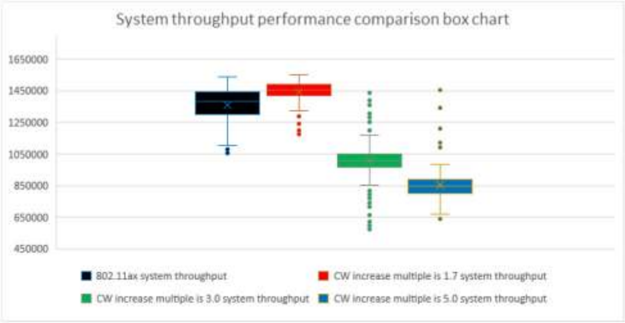

Fig. 14. Statistical performance of system throughput.

under variable CW increase multiple. The figure shows that the prediction result of the neural network can accurately reflect the change in system throughput. When the CW increase multiple is 2.5–6.5 times, the predicted data result is higher than the original data, and the maximum deviation is less than 6%, which is a normal phenomenon in the data prediction process.

Fig. 11 shows the comparison between the original data of system delay and the predicted data of neural network under the variable CW increase multiple. The figure shows that the prediction curve of the network basically coincides with the curve of the original data. This finding indicates that this network can predict the change in system delay well.

Fig. 12 shows the comparison between the original data of the packet retransmission rate and the neural network forecast data under the variable CW increase multiple. The data prediction curve only fluctuates at CW increase multiples of 1.8–1.9 and 2.5–3.0, and the floating ranges are less than 5%, which is a normal floating range. This result indicates that this network can predict the packet retransmission rate effectively.

Fig. 13 shows a comparison chart of system throughput performance. The figure shows that in the simulation time of 200 s, the system throughput performance is the highest when the CW increase multiple is 1.7 times, and compared with the 802.11ax scheme, the throughput performance is improved by approximately 7%. In scenes that require high system throughput, such as industrial video data transmission and data backup, the proposed optimization algorithm can be used to select the CW increase multiple that maximizes the system throughput to improve the network throughput performance, thus, satisfy meet the needs of these scenes.

Fig. 14 shows a box chart of the system throughput performance comparison. The figure shows that when the CW increase multiple is 1.7 times, the system throughput is the highest and the floating range of the throughput is the smallest. Compared with the 802.11ax scheme, the scheme with a CW increase at multiple of 1.7 times is more suitable for scenes with higher throughput performance requirements.

Fig. 15 shows a comparison chart of system delay performance. The figure shows that within 200 s of simulation time, the system delay of the 802.11ax scheme is higher. However, the system delay is the lowest when the CW increase multiple is five times. In scenes that require high system delay performance, such as mobile access scenarios and voice calls, the proposed optimization algorithm can be used to select a suitable CW increase multiple to improve network delay performance. For example, the CW increase multiple can be set to three times to satisfy the requirements of such scenes. The CW increase factor is selected to be three times given that

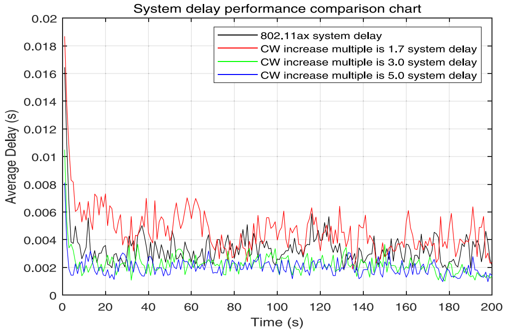

Fig. 15. Average transmission delay.

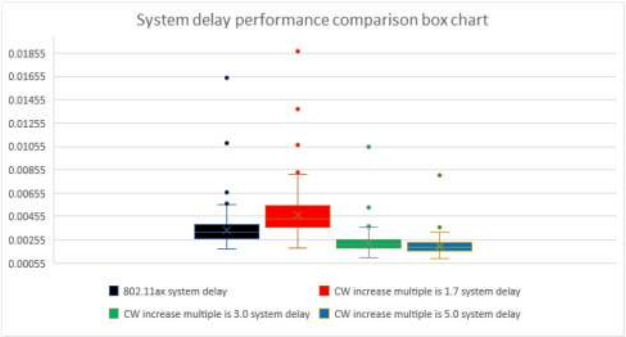

Fig. 16. Statistical performance of average transmission delay.

Figs. 14 and 16 show that the system delay performance difference between the CW increase factor of three times and five times is small. However, at three times, the system throughput performance is approximately 17% higher than at five times.

Fig. 16 shows a box chart of system delay performance comparison. The figure shows that when the CW increase multiple is five times, the system delay is the lowest and the floating range of the delay is the smallest. The scheme with a larger CW increase multiple (such as three times or five times) is more suitable for scenes with higher delay performance requirements than the 802.11ax scheme.

Fig. 17 shows a comparison chart of system packet retransmission rate performance. The figure shows that within 200 s of simulation time, the data packet retransmission rate of the 802.11ax scheme is higher, but the data packet retransmission rate is the lowest when the CW increase multiple is 5. In scenes that require high performance for packet retransmission rates, such as industrial sensor communication scenarios, the CW increase multiple of three times can be selected to improve the performance of network packet retransmission rates to satisfy the needs of these scenes.

Fig. 18 shows a box chart of system data packet retransmission rate performance comparison. The figure shows that the system data packet retransmission rate is the lowest when the CW increase multiple is five times. Compared with the 802.11ax scheme, the scheme with a larger CW increase multiple (three times or five times) is more suitable for scenes with higher performance requirements on the packet retransmission rate.

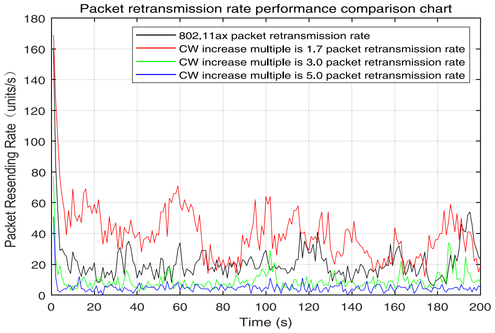

Fig. 17. Packet retransmission rate.

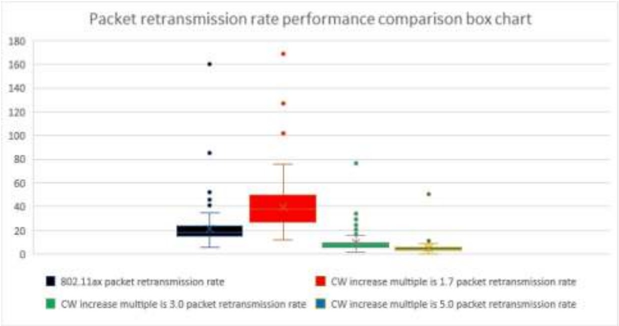

Fig. 18. Statistical performance of packet retransmission rate.

VI. Conclusion

The CSMA/CA protocol has been widely used in the field of Wi-Fi-enabled IoT systems. However, the BEB strategy limits the optimization of network performance. Deep learning algorithms are used to establish a model between the backoff window increase multiple and the system throughput, average transmission delay, and data packet retransmission rate. This model can effectively predict the system performance of different increase multiples of the backoff window. On the basis of the prediction results, the increase multiples of the backoff window are selected to optimize the system performance and used the CSMA/CA protocol to improve the performance of wireless communications. We use the Bayesian statistical regularization algorithm to train the network and introduce a dynamically changing learning rate to enhance the generalization ability of the network. At the same time, a threshold is set to detect algorithm generalization ability in the network. If the number of consecutive training of the algorithm exceeds this threshold and the training error does not decrease but increases, then the training is forced to end. These measures enable this network to obtain a strong generalization ability that can be used in different applications without repeated training. The experimental simulation results showed that the performance of system throughput, system delay, and packet retransmission rate is improved using the predicted optimal backoff window increase multiple. A suitable CW increasing multiple can be selected from the model prediction results in accordance with the system performance priority level to satisfy the performance tradeoffs among the parameters of the system performance index. Thus, a reasonable improvement of the system performance parameters is achieved. Moreover, this algorithm does not require multiple training for various application scenarios. Our future works will investigate the Wi-Fi 6 performance improvement with multidimensional features introduced, e.g., channel, coding, and interleaving.

References

[1] C. Chen, L. Liu, T. Qiu, K. Yang, F. Gong, and H. Song, "ASGR: An artificial spider-web-based geographic routing in heterogeneous vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1604–1620, May 2019.

[2] J. J. Cheng, G. Y. Yuan, M. C. Zhou, S. C. Gao, Z. H. Huang, and C. Liu, "A connectivity-prediction-based dynamic clustering model for VANET in an urban scene," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8410–8418, Sep. 2020.

[3] S. Li, Q. Ni, Y. Sun, G. Min, and S. Al-Rubaye, "Energy-efficient resource allocation for industrial cyber-physical IoT systems in 5G era," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2618–2628, Jun. 2018.

[4] S. Wan, X. Xu, T. Wang, and Z. Gu, "An intelligent video analysis method for abnormal event detection in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, early access, Sep. 9, 2020, doi: 10.1109/TITS.2020.3017505.

[5] J. Hu, C. Chen, T. Qiu, and Q. Pei, "Regional-centralized content dissemination for eV2X services in 5G mmwave-enabled IoV," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7234–7249, Aug. 2020.

[6] S. Jacob *et al.*, "A novel spectrum sharing scheme using dynamic long short-term memory with CP-OFDMA in 5G networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 926–934, Sep. 2020.

[7] C. Chen, X. Liu, T. Qiu, and A. K. Sangaiah, "A short-term traffic prediction model in the vehicular cyber–physical systems," *Future Gener. Comput. Syst.*, vol. 105, pp. 894–903, Apr. 2020.

[8] B. Wang, Y. Sun, S. Li, and Q. Cao, "Hierarchical matching with peer effect for low-latency and high-reliable caching in social IoT," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1193–1209, Feb. 2019.

[9] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1079–1092, Feb. 2019.

[10] C. Chen, J. Hu, T. Qiu, M. Atiquzzaman, and Z. Ren, "CVCG: Cooperative V2V-aided transmission scheme based on coalitional game for popular content distribution in vehicular ad-hoc networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 12, pp. 2811–2828, Dec. 2019.

[11] M. R. Khosravi and S. Samadi, "Reliable data aggregation in Internet of ViSAR vehicles using chained dual-phase adaptive interpolation and data embedding," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2603–2610, Apr. 2020.

[12] X. Li, M. Huang, Y. Liu, V. G. Menon, A. Paul, and Z. Ding, "I/Q imbalance aware nonlinear wireless-powered relaying of B5G networks: Security and reliability analysis," 2020. [Online]. Available: arXiv:2006.03902.

[13] D. G. Filoso, R. Kubo, K. Hara, S. Tamaki, K. Minami, and K. Tsuji, "Proportional-based resource allocation control with QoS adaptation for IEEE 802.11ax," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, 2020, pp. 1–6.

[14] A. F. Rochim, B. Harijadi, Y. P. Purbanugraha, S. Fuad, and K. A. Nugroho, "Performance comparison of wireless protocol IEEE 802.11ax vs 802.11ac," in *Proc. Int. Conf. Smart Technol. Appl. (ICoSTA)*, Surabaya, Indonesia, 2020, pp. 1–5.

[15] E. Lu *et al.*, "10.4 a 4×4 dual-band dual-concurrent WiFi 802.11ax transceiver with integrated LNA, PA and T/R switch achieving +20dBm 1024-QAM MCS11 pout and –43dB EVM floor in 55nm CMOS," in *Proc. IEEE Int. Solid- State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2020, pp. 178–180.

[16] S. Lin, "Research on efficient MAC layer access mechanism in WLAN," Ph.D. dissertation, Dept. Inf. Telecommun. Eng., Beijing Univ. Posts Telecommun., Beijing, China, 2019.

[17] W.-Z. Wang, J.-L. Zhou, Z. Long, and P.-C. Luo, "Message-streams oriented hybrid slot allocation protocol for tactical data link system," in *Proc. 7th Annu. Conf. Commun. Netw. Serv. Res. (CNSR)*, Moncton, NB, Canada, 2009, pp. 11–13.

[18] R. Ali, N. Shahin, R. Bajracharya, B.-S. Kim, and S. W. Kim, "A self-scrutinized backoff mechanism for IEEE 802.11ax in 5G unlicensed networks," *Sustainability*, vol. 10, no. 4, p. 1201, 2018.

[19] N. Shahin, R. Ali, S. W. Kim, and Y.-T. Kim, "Adaptively scaled back-off (ASB) mechanism for enhanced performance of CSMA/CA in IEEE 802.11ax high efficiency WLAN," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, Taipei, Taiwan, 2018, pp. 1–5.

[20] M. Hasan, "Performance enhancement of the EDCA protocol for the channel access in next generation IEEE 802.11 AX WLAN," M.S. thesis, Dept. Inst. Inf. Commun. Technol., Bangladesh Univ. Eng. Technol., Dhaka, Bangladesh, 2019.

[21] T. Nabetani, N. Madhavan, H. Mori, and T. Aoki, "A novel low-overhead channel sounding protocol for downlink multi-user MIMO in IEEE 802.11ax WLAN," *IEICE Trans. Commun.*, vol. E101.B, no. 3, pp. 924–932, 2018.

[22] C. Chen, C. Wang, T. Qiu, M. Atiquzzaman, and D. O. Wu, "Caching in vehicular named data networking: Architecture, schemes and future directions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2378–2407, 4th Quart., 2020.

[23] M. Abbasi, A. Shokrollahi, M. R. Khosravi, and V. G. Menon, "High-performance flow classification using hybrid clusters in software defined mobile edge computing," *Comput. Commun.*, vol. 160, pp. 643–660, Jul. 2020.

[24] K. Wang, M. Jin, L. Kang, H. Zhang, and L. Zhang, "Simulation analysis of TDMA/CSMA hybrid protocol based on MATLAB," *Fire Command Control*, vol. 42, no. 2, pp. 163–167, 2017.

[25] H. Liu, "Research on access algorithm of ad hoc network based on backoff mechanism," Ph.D. dissertation, Dept. Electr. Inf. Eng., Harbin Inst. Technol., Harbin, China, 2016.

[26] X. Xu, "Research and algorithm improvement of MAC layer protocol in ad hoc network," Ph.D. dissertation, Dept. Inf. Telecommun. Eng., Univ. Electron. Sci. Technol. China, Chengdu, China, 2017.

[27] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun. 12th Annu. Joint Conf. IEEE Comput. Commun. Soc. (Cat. No.01CH37213)*, Anchorage, AK, USA, 2001, pp. 189–198.

[28] J. Shi, J. Liu, and H. Qin, "A priority based CSMA/CA optimization algorithm in ZigBee network," *J. Sens. Technol.*, vol. 31, no. 06, pp. 104–110, 2018.

[29] J. Wang, D. Li, and S. Liu, "802.11DCF backoff algorithm improvement and simulation," *Electron. Design Eng.*, vol. 25, no. 10, pp. 148–153, 2017.

[30] R. Ali, N. Shahin, Y.-T. Kim, B.-S. Kim, and S. W. Kim, "Channel observation-based scaled backoff mechanism for high-efficiency WLANs," *Electron. Lett.*, vol. 54, no. 10, pp. 663–665, May 2018.

[31] A. H. Y. Abyaneh, M. Hirzallah, and M. Krunz, "Intelligent-CW: AI-based framework for controlling contention window in WLANs," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Newark, NJ, USA, 2019, pp. 1–10.

[32] R. Ali, A. Nauman, Y. Zikria, B.-S. Kim, and S. W. Kim, "Performance optimization of QoS-supported dense WLANs using machine-learning-enabled enhanced distributed channel access (MEDCA) mechanism," *Neural Comput. Appl.*, vol. 7, no. 1, pp. 1–9, 2019.

[33] R. Ali, N. Shahin, Y. B. Zikria, B. Kim, and S. W. Kim, "Deep reinforcement learning paradigm for performance optimization of channel observation-based MAC protocols in dense WLANs," *IEEE Access*, vol. 7, pp. 3500–3511, 2019.

[34] X. Liu, *MATLAB/Simulink*, Publ. House Electron. Ind., Beijing, China, 2013.

[35] T. Zhou, C. Wu, and Y. Zhang, "Simulation model of Nakagami-m complex fading channel," *Electron. Meas. Technol.*, vol. 10, no. 13, pp. 36–39, 2019.

[36] L. Jianwei, Z. Huidan, L. Xionglin, and X. Jun, "Research progress of batch normalization and related algorithms in deep learning," *Acta Autom. Sinica*, vol. 46, no. 6, pp. 1090–1120, 2020.

[37] K. Lee, S. H. Sung, D. H. Kim, and S.-H. Park, "Verification of normalization effects through comparison of CNN models," in *Proc. Int. Conf. Multimedia Anal. Pattern Recognit. (MAPR)*, Ho Chi Minh City, Vietnam, 2019, pp. 1–5.

[38] T. Kwon, "Average data rate analysis for hierachical cell structure under Nakagami-m fading channel with a two-layer feed-forward neural network," in *Proc. Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, Barcelona, Spain, 2019, pp. 1–4.

[39] J. Guan, R. Lai, A. Xiong, Z. Liu, and L. Gu, "Fixed pattern noise reduction for infrared images based on cascade residual attention CNN," *Neurocomputing*, vol. 377, pp. 301–313, Feb. 2020.

[40] J. Guan, R. Lai, A. Xiong, Z. Liu, and L. Gu, "Fixed pattern noise reduction for infrared images based on cascade residual attention CNN," *Neurocomputing*, vol. 377, no. 15, pp. 301–313, 2020.

Chen Chen (Senior Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees in telecommunication from Xidian University, Xi'an, China, in 2000, 2006, and 2008, respectively.

He is currently a Professor with the Department of telecommunication and a Member of the State Key Laboratory of Integrated Service Networks, Xidian University, where he is also the Director of Xi'an Key Laboratory of Mobile Edge Computing and Security and the Intelligent Transportation Research Laboratory. He was a Visiting Professor with the Department of EECS, University of Tennessee, Knoxville, TN, USA, and the Department of CS, University of California at Santa Barbara, Santa Barbara, CA, USA. He serves as the general chair, a PC chair, the workshop chair, or a TPC member of a number of conferences. He has authored/coauthored two books, over 100 scientific papers in international journals and conference proceedings. He has contributed to the development of five copyrighted software systems and invented over 100 patents.

Dr. Chen is also a Senior Member of China Computer Federation and a member of ACM and Chinese Institute of Electronics.

Junchao Li received the B.Eng. degree in communication engineering from Xidian University, Xi'an, China, in 2017, where he is currently pursuing the master's degree.

His major is electronic and communications engineering. His research interests include embedded development and wireless *ad hoc* networks.

Venki Balasubramaniam (Member, IEEE) received the Ph.D. degree in body area wireless sensor network for remote healthcare monitoring applications from the University of Technology at Sydney, Ultimo, NSW, Australia.

He is the Pioneer in building (pilot) remote healthcare monitoring application for pregnant women with the New South Wales Healthcare Department, St. Leonards, NSW. His research establishes a dependability measure to evaluate rHMA that uses BAWSN. He contributed immensely to eResearch software research and development that uses cloud-based infrastructure and a Core Member for the project sponsored by Nectar Australian research cloud provider. He contributed heavily in the field of healthcare informatics, sensor networks, and cloud computing. He also founded Anidra Tech Ventures Pty Ltd., a smart remote patient monitoring company. His research opens up a new research area in measuring time-critical applications.

Yongqiang Wu received the bachelor's degree from the Secondary Specialized School, Industrial Enterprise Automation, Wenzhou Mechanical Industry School, Zhejiang, China, in 1985.

Since 1997, he has been Founded Zhejiang Wellsun Intelligent Technology Company, Ltd. Zhejiang. He received the Senior Engineer qualification. His research interests include wireless *ad hoc* networks, smart grid, and smart fire system.

Yuru Zhang received the B.Eng. degree in communication engineering from Lanzhou Jiaotong University, Lanzhou, China, in 2019. She is currently pursuing the master's degree with Xidian University, Xi'an, China.

Her major is transportation information engineering and control. Her research interests include Internet of Vehicle and wireless communication.

Shaohua Wan (Senior Member, IEEE) received the joint Ph.D. degrees from the School of Computer, Wuhan University, Wuhan, China, and the Department of Electrical Engineering and Computer Science, Northwestern University, St. Evanston, IL, USA, in 2010.

Since 2015, he has been holding a Postdoctoral position with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan. From 2016 to 2017, he was a Visiting Professor with the Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany. He is currently an Associate Professor with the School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan. He has an authored of over 90 peer-reviewed research papers and books. His main research interests include deep learning for Internet of Things and edge computing.