International Journal of Innovative Computing, Information and Control
Volume **20**, Number **6**, December **2024**

 ISSN 1349-4198
pp. **1749–1763**

INTERNET OF THINGS EDGE DATA MINING TECHNOLOGY BASED ON CLOUD COMPUTING MODEL

Ning Hu

School of Economics and Management
Kaifeng Vocational College of Culture and Arts
Middle Section of Dongjing Avenue, Longting District, Kaifeng 475000, P. R. China
huning@kfwyxy.edu.cn

Received November 2023; revised March 2024

Abstract. *Due to the limited computing resources of Internet of Things edge devices, cloud and edge computing can work together to improve the way data is processed and services are delivered. Cloud computing focuses on providing centralized resources, while edge computing pushes data processing to the edge of the network, reducing latency. To this end, the research proposes to use edge computing for data processing on the cloud computing platform to reduce the pressure of network bandwidth. It utilizes the addition of early exit branch neural networks to process data on edge devices. Simultaneously through a linear regression model, the optimal branch exit point is determined based on network latency and device load for enhancing the efficiency and accuracy of data classification. The results showed that the training accuracy of AlexNet and Resnet networks with branches also increased with the increase of training times. The training accuracy of Layer 2 was maintained within the range of 0.98-1.0 and 0.99-1.0, respectively. When the latency was 1.25 ms, the classification accuracy of the branch AlexNet network differed the most from that of the unbranched AlexNet network by 15%. When the delay was 3.5 ms, the classification accuracy of the branch Resnet network was 4.1% higher than that of the unbranched Resnet network. The classification inference performance of the research model is the best, with a performance detection value of 0.92. The research method can serve as a new model for addressing the requirements of the Internet of Things and localized computing, and has important reference value in reducing the latency and data transmission bandwidth of computing systems.*

Keywords: Cloud computing, Edge computing, Internet of Things, Data mining, Deep neural network

1. **Introduction.** The Internet of Things (IoT) refers to the connection between items and the Internet, enabling automatic identification of each other between each item. It can communicate with the other party to a certain extent, and even achieve independent decision-making of the device through Internet technology [1]. IoT is built on top of the traditional Internet, achieving interaction between machines and humans. As the boost of the Internet, the coverage of the IoT has become increasingly rich. Nowadays, IoT has developed into various aspects such as healthcare and daily life. IoT relies on sensors to achieve communication and communication between devices; Relying on sensors to achieve communication between machines can produce abundant data. As the scope of IoT continues to expand, the sensors in devices are becoming increasingly complex [2]. Providing better decision-making for the intelligence of IoT has become a major issue faced by scholars. Data mining (DM) is a process of finding unknown information and knowledge from abundant data. It utilizes various methods such as statistics and artificial intelligence to discover hidden, unknown, and practical knowledge and patterns from vast,

DOI: 10.24507/ijicic.20.06.1749

complex, incomplete, random, and fuzzy actual data. If DM is combined with the IoT, it can effectively help the IoT extract effective information from big data and use this information in device decision-making. However, with the rapid growth in the amount of IoT data, it becomes particularly important to use DM techniques to extract valuable information from it. This helps IoT provide smarter services [3]. However, because image data processing requires intensive computing, the data is usually sent to the cloud computing server for processing. However, this method is limited by network delay and bandwidth, so it is difficult to meet the demand for immediacy, and may increase the risk of privacy disclosure [4]. Therefore, it is of great research value to transfer part of DM tasks to IoT edge devices to realize real-time data processing in edge computing (EC) environment [5]. The innovation of the research lies in the introduction of DM technology in the IoT. Meanwhile, the task of DM is put on the edge of the IoT EC server, and EC is used to mine the big data of the IoT. This study aims to improve the efficiency and reliability of the IoT in serving users, and to more scientifically and quickly mine valuable knowledge from massive data. The main contributions of the research include the following aspects. Firstly, EC is used to analyze the data, which relieves the pressure of network bandwidth. Secondly, the use of convolutional neural network (CNN) for image classification is discussed, and neural network performs effective data processing on edge devices by exiting branches in advance. Then, a linear regression model is applied to determine the optimal Branch Exit Points (BEP) based on network latency and device health, thus improving the speed and accuracy of data processing in cloud computing and EC collaboration. Finally, the CNN simulation experiment is carried out using Chainer platform to verify the feasibility and effectiveness of the proposed method. The research mainly includes five parts. Part 1 introduces the background and significance of DM related to the IoT. Part 2 presents a summary of the overview of IoT DM. Part 3 proposes the research methodology, mainly divided into two sections. In Section 3.1, a collaborative method integrating EC and cloud computing is proposed. In Section 3.2, a cloud computing platform-based IoT edge DM model is constructed. Part 4 is about verifying the effectiveness of the research model. Part 5 gives a summary of the latest research methods and an analysis of the experimental results. Meanwhile, the shortcomings of research methods and future research directions are proposed.

2. **Related Works.** As the boost of the IoT, the data that needs to be processed from IoT devices has become increasingly large and complex. DM technology is a common big data processing method, and DM plays an important role in the IoTs. Xie et al. constructed a social e-commerce information model for enriching the theoretical system of user DM on e-commerce platforms in the big data of the IoT, and used social network analysis to analyze the e-commerce user network. The results showcased that new technology was a key factor affecting the initial information adoption of social e-commerce users, with a heart rate of 5.25 [5]. In view of the shortcomings such as the efficiency of DM technology in the IoTs, many scholars have also proposed improvement methods. Wu et al. proposed to introduce a mining algorithm framework for processing large datasets into existing algorithms to address the issue of efficient sequence mining algorithms in IoT that can only run in a standalone environment and suffer from efficiency degradation when faced with massive datasets. The results showed that the proposed algorithm had significant advantages in efficient probability sequence mining in large datasets [6]. Li proposed a DM algorithm on the ground of multi-tree aggregation to optimize wireless communication technology in IoT, and optimized the basic wireless communication of IoT. The results showed that the proposed algorithm performed well in similar DM technologies, and its optimization effect on IoT-based wireless communication also met the

experimental design requirements [7]. After the technology of the IoTs has been optimized, some scholars have proposed to improve its security research. Azrour et al. proposed an enhanced authentication protocol in the IoT to improve user data security, and used it to encrypt and protect user data. The results showcased that the proposed method could provide better protection for user data privacy [8]. Chen et al. proposed a blockchain-based IoT data trading method to improve the security and efficiency of data trading in IoT. This method was on the ground of an iterative bidirectional auction mechanism, which can gradually extract individual information instead of extracting it all at once. The outcomes showcased that the algorithm could markedly enhance the security of user information and shorten the latency of information transactions [9].

EC is one of the IoT EC models and a common way for IoT to process massive data. Xue et al. presented a hybrid algorithm for enhancing the system service performance of IoT, which solved the resource scheduling problem with both parallelism and subtask dependency in IoT. The results showed that the proposed algorithm converged faster than traditional methods and had better optimization effects on resource scheduling [10]. To improve the security of vehicle EC networks, Bai et al. proposed a defense algorithm that used edge nodes to detect the international ISDN number of conflicting mobile users. The results showed that the algorithm could markedly enhance the security of vehicle EC network [11]. To make satellite networks serve as computing servers for IoT users in remote areas, Gao et al. presented a potential game method for virtual network function layout in satellite network EC. The results showed that the algorithm could markedly address the problem of virtual network function layout in satellite EC [12]. To reduce the computing load of EC server in the concrete bridge crack detection system, Yang et al. presented a concrete crack segmentation network on the ground of UAV EC. The outcomes showcased that the method was higher in accuracy and universality than the current edge detection and semantic segmentation methods [13]. Zheng et al. proposed a distributed hierarchical tensor deep optimization algorithm to reduce bandwidth consumption during federated learning system updates. This algorithm compressed model parameters from high-dimensional to low-dimensional, and utilized backpropagation update for reducing the memory requirement for system updates. The results showed that the algorithm could markedly reduce the communication bandwidth burden and resource requirements for edge node computing [14].

In summary, it is feasible to add DM to the EC server model of IoT and process the massive data in IoT. However, conducting massive DM requires a high demand for computing resources. It can markedly reduce the computing load of EC servers. Therefore, this research uses EC to deal with massive DM in IoT to improve task scheduling decisions of IoT devices and improve information security of IoT users. All formula symbols are explained in Table 1.

3. Edge DM Method of IoT on the Ground of EC Model.

In this study, a multi-agent model on the ground of cloud computing and EC is used to add early exit branches to the CNN used for image data classification. Then it uses a linear regression model to select appropriate exit branch points on the ground of network latency and device load. By branching points, the CNN is divided into shallow parts containing inputs and deep parts containing outputs in a hierarchical structure. On edge devices, the shallow part of the neural network (NN) is deployed to establish edge agents. It deploys deep layers of NNs on cloud server devices to establish cloud agents, and constructs multi-agent systems through edge cloud collaboration.

TABLE 1. Formula symbols and their meanings

Formula symbol	Implication	Formula symbol	Implication
x	Input sample	a	Sample attribute
y	True label	$GainRatio(D,a)$	Information gain rate
S	Tag set	$IV(a)$	Punitiveness
$\hat{y}$	Predictive output	$f(x_i)$	Linear regression model
z	Network layer output	x_{in}	Independent variable
θ	Parameter set	y_i	Dependent variable
f_{exitn}	Branch exit point output	w, b	Regression coefficient
ω_n	The weights of each branch model	B	Network bandwidth
L_n	Exit point number	ET_i	Run time on edge devices
$Ent(D)$	Information entropy	CT_i	Uptime on the cloud server
α	Negative gradient	$X_{M\times N}$	Enter the image block data set
C	Covariance matrix	X_{PCA_whiten}	PCA whitening value
ε	Safety factor	U	Vector quantity

3.1. **A collaborative approach integrating cloud computing and EC.** EC is for handling the data generated by embedded devices of the IoT, and its actual deployment has the characteristics of natural distribution [15]. It conducts centralized model training of deep neural network (DNN) models in the cloud and distributes the models to edge nodes. Edge nodes perform inference on the ground of DNN models to achieve distributed intelligence. It deploys the shallow of the DNN with an input layer into edge devices to form an edge agent. It deploys deep layers into resource rich cloud servers, forming cloud-based intelligent agents. The DNN structure on the ground of distributed computing hierarchy for hybrid cloud and edge is shown in Figure 1.

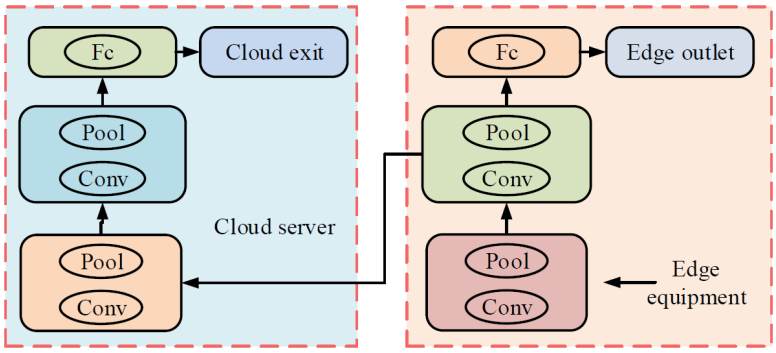


FIGURE 1. Hierarchical distributed DNN

In Figure 1, in cloud computing, "Fc" usually refers to an implementation of "Function as a Service", and may also refer specifically to the functional computing services of some cloud service platforms. The "Fc" service is usually billed according to the number of times the function is executed and the execution time. The shallow of the edge device can perform preliminary classification and feature extraction on the data. If the shallow part does not perform well in data classification, the extracted data features are handed over to the deep to achieve more exact data classification. Due to the fact that the number of neurons in the middle of the NN can be lower than that in the input layer, the output

of the middle is designed to be lower than the input. This is for decreasing the network communication required in terminal devices and the cloud. Since the collected raw data is not uploaded, it can protect user privacy and data security. This study adds exit branches at certain locations throughout the entire network to modify the standard deep network structure [16].

This study trains NNs with branch structures through addressing the joint optimization issue of weighted sum of loss functions associated with BEP. It is first defined as an input sample, which is the true label of the sample. The expression of the loss function is shown in Equation (1).

$$L_n(\hat{y}, y; \theta) = -\frac{1}{|s|} \sum_{s \in S} y_s \log \hat{y}_s \tag{1}$$

In Equation (1), S serves as the set of labels for all possible samples. $\hat{y}$ serves as the predicted output of the output sample, and its expression is shown in Equation (2).

$$\hat{y} = soft\max(z) = \frac{\exp(z)}{\sum_{s \in S} \exp(z_s)} \tag{2}$$

In Equation (2), z represents the output of the network layer, and its expression is shown in Equation (3).

$$z = f_{exitn}(x; \theta) \tag{3}$$

In Equation (3), θ represents the parameter set of a branch network layer from the entry to the exit. f_{exitn} represents the output of the n-th model BEP of the branch network model. This study adopts a special strategy for branch network models. Among them, centralized learning is used to handle environmental instability and consider the joint action effects of multi-agent systems. The loss value of each exit point needs to be multiplied by its weight to achieve joint optimization. The final loss function is showcased in Equation (4).

$$L(\hat{y}, y; \theta) = \sum_{n=1}^{N} \omega_n L_n(\hat{y}, y; \theta) \tag{4}$$

In Equation (4), ω_n serves as the weight of each branch model. L_n serves as the quantity of all exit points. This algorithm includes two steps. During the feedforward, the error of the NN can be obtained. In feedback, the error is transmitted back by the network. It updates the weight. This study uses the Adam optimization algorithm for model training, and uses the softmax function to normalize the training output, generating all class probability sets between 0 and 1 [17]. Information entropy is the most commonly used indicator for measuring the purity of sample sets. Assuming that the proportion of Class i samples in sample set D is P_i, then the calculation expression of information entropy is shown in Equation (5).

$$Ent(D) = -\sum_{i=1}^{|y|} P_i \log_2 P_i \tag{5}$$

In Equation (5), the smaller $Ent(D)$, the higher the purity of the set D. The prediction probability of each class label is defined as y_s, and all possible sets of class labels are defined as S. The sample output information entropy (OIE) of the exit point is defined as Equation (6).

$$entropy(y) = \sum_{s \in S} y_s \log y_s \tag{6}$$

The smaller the OIE of the test sample x at the BEP, the higher the confidence of the classifier at the BEP in the correctly labeled prediction results, and the greater the

likelihood of the sample being prematurely exited from the network. The calculation expression of information gain is shown in Equation (7).

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D_v|}{D} Ent(D_v) \tag{7}$$

In Equation (7), a represents a certain attribute of the sample. The relevant calculation is showcased in Equation (8).

$$\begin{cases} GainRatio(D,a) = \dfrac{Gain(D,a)}{IV(a)} \\ IV(a) = -\displaystyle\sum_{v=1}^{V} \frac{|D_v|}{D} \log_2 \frac{|D_v|}{D} \end{cases} \tag{8}$$

In Equation (8), $IV(a)$ represents punitive nature. In NNs, the larger the eigenvalues x, the greater the $Wx+b$. Therefore, when outputting the activation function, it will result in a small change in the numerical value of the corresponding position, which is easy to fit and has poor results. ReLU has a significant accelerating effect on the convergence of random gradient descent [18]. To address the problem of hard saturation in the ReLU function at $x < 0$, activation functions such as Leaky-_ReLU and ELU are performed on the basis of ReLU. The expression for Leaky-ReLU is shown in Equation (9).

$$\phi(x) = \max(\alpha x, x) \tag{9}$$

In Equation (9), α represents a very small negative gradient value. The mathematical expression of the ELU function is shown in Equation (10).

$$\phi(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \le 0 \end{cases} \tag{10}$$

ReLU can maintain the gradient without decay at time $x > 0$, thereby alleviating the problem of gradient disappearance. Therefore, it is necessary to perform de-averaging on the graph to remove the average brightness value of the image. The process diagram of image averaging is shown in Figure 2 [19].

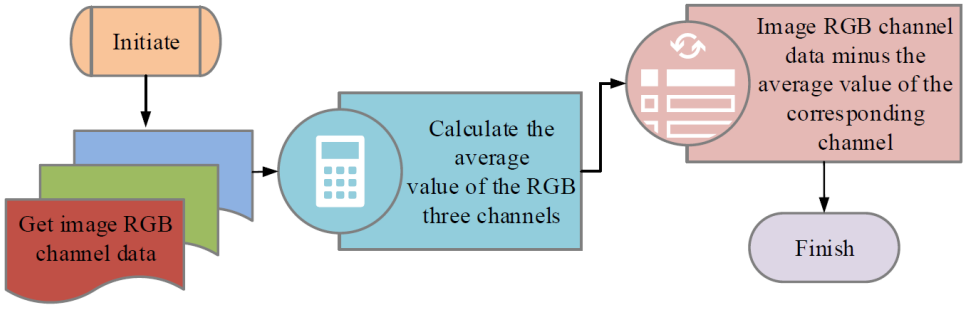


FIGURE 2. Flow chart of image averaging

In Figure 2, the process of de-averaging the image first involves obtaining RGB channel data of the image and calculating the average values of the three channels. By subtracting the average value of the corresponding channel from the data of the RGB channel of the image, a de-averaged image can be obtained.

3.2. **Construction of IoT edge DM model based on EC.** A multi-agent system is a collection of multiple intelligent agents, and in an intelligent IoT environment, each device can be regarded as an intelligent agent. The branch NN model is used to various devices on the ground of the horizontal splitting model of each branch, forming a cooperative multi-agent system. Due to the different computing functions and composition structures of each layer in the NN, there are differences in the calculation delay and input/output data size between different layers. Therefore, this study establishes linear regression models for various layer types on the ground of the parameters of different layers to predict layer delays on the ground of this configuration. If a dataset $\{y_i, x_{i1}, x_{i2}, \ldots, x_{im}\}_{i=0}^{n}$ containing n sets of data is given, assuming an implicit linear relation in the independent variable and the dependent variable, a linear regression model as shown in Equation (11) can be constructed.

$$f(x_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in} + b_i, \quad i = 1, 2, \ldots, n \tag{11}$$

In Equation (11), x_{in} serves as the independent variable. w, b represent the regression coefficients of linear equations. This linear regression algorithm learns the appropriate w, b on the ground of fitting a given dataset, thus making it as close as possible to y_i. Multi-agent systems are autonomous and interacting entities that share the same environment. It utilizes sensors to sense and actuators to act. The optimal partitioning point of a DNN depends on its topological structure. Under a specific network bandwidth B, the NN model possesses N partitioning points. The relevant expression is showcased in Equation (12).

$$T_i = \sum_{i=1}^{j} ET_i + \sum_{i=j+1}^{N} CT_i + \frac{o_j}{B} \tag{12}$$

In Equation (12), ET_i represents the running time of layer i. CT_i represents the running time of layer i on the cloud server. o_j serves as the output size of j.

Principal component analysis (PCA) can preserve the main feature information for dimensionality reduction of data. PCA whitening refers to the use of feature value factor scaling to minimize the correlation between different features and ensure that all feature variances are equal after PCA dimensionality reduction of input data [20]. Assuming that the input image block data set is $X_{M\times N}$, its covariance matrix is shown in Equation (13).

$$C = \mathrm{cov}(X_{M\times N}) \tag{13}$$

By eigenvalue decomposition of Equation (13), Equation (14) can be obtained.

$$[V, D] = eig(C) \tag{14}$$

The calculation expression of PCA whitening is shown in Equation (15).

$$X_{PCA\_whiten} = \frac{(X_{M\times N} - mean(X_{M\times N}) \cdot U)}{\sqrt{diag(D) + \varepsilon}} \tag{15}$$

There is a possibility of severe noise reduction during whitening operations. In the PCA calculation method, a constant is added to the denominator [21]. Safety factor ε is used to prevent the occurrence of dividing by 0. It is generally 10^{-5}. However, the noise portion in the data may become larger due to whitening operations. Therefore, this study can alleviate this problem to some extent by appropriately increasing the safety factor in the denominator. ZCA whitening is a rotation operation on the ground of PCA whitening, and its calculation formula is as shown in Equation (16).

$$X_{ZCA\_whiten} = U \cdot X_{PCA\_whiten} \tag{16}$$

ZCA whitening removes many correlations between features through PCA and calculates the unit deviation of input features [22]. It then performs a rotation operation to obtain the result of ZCA, preserving all dimensions of the data without performing a dimensionality reduction operation. The image whitening processing steps are shown in Figure 3.

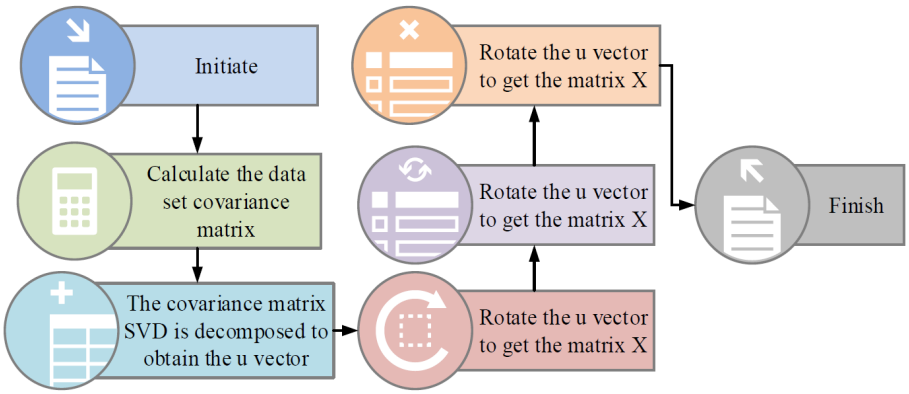


FIGURE 3. Image whitening process

The operation of NN models is mainly divided into two processes, namely the training and optimization process and the classification process. The training and validation sets add newly collected image data to the publicly available dataset already used for model training. After scrambling the new dataset, the already trained model is fine-tuned to continuously optimize and update it to become more robust. The relevant details of the NN operation are indicated in Figure 4.

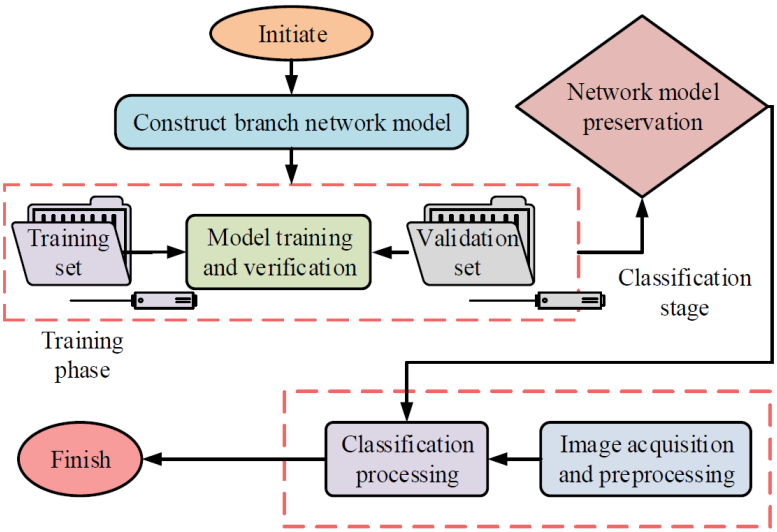


FIGURE 4. Operation diagram of NN

A branch NN extends a branch at a specific intermediate layer position in a general DNN as an early output node [23]. This allows the model to output from the branches in advance during the forward propagation process, without the need for deep network calculations, or to output from the original deeper output nodes, achieving better results. The basic structure of the network is shown in Figure 5.

In Figure 5, the network outputs Output_2 from shallow branches and Output_1 from the backbone network. For the input sample, both outputs are the results of network

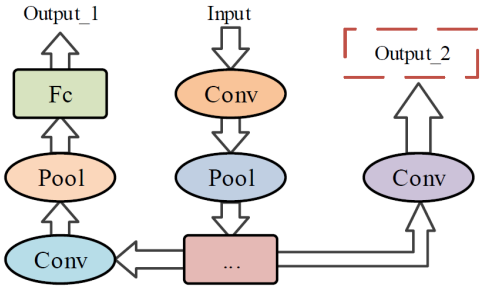


FIGURE 5. Branch NN

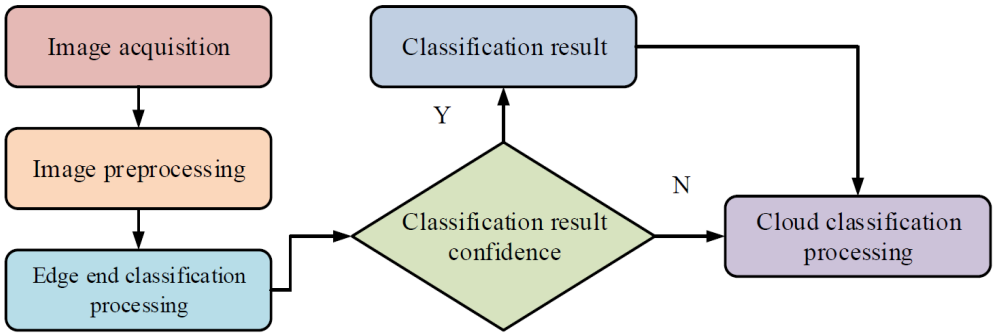


FIGURE 6. Overall framework of multi-agent system based on cloud computing and EC

prediction. Therefore, when calculating the loss, there will be both the loss L1 of Output_1 and the loss L2 of Output_2, with the total loss being the weighted sum of L1 and L2. The optimization of network parameters still uses backpropagation algorithm, which is theoretically equivalent to optimizing two different networks simultaneously in one backpropagation. The NN layer before the BEP is deployed in edge devices, and most of the data is classified and processed locally [24,25]. The overall framework of the multi-agent system on the ground of cloud computing and EC is shown in Figure 6.

In Figure 6, the research on EC-based IoT edge DM mainly needs to implement three modules, namely data collection, data pre-processing, and data classification processing module. The image data collected through sensors is classified and recognized using CNNs deployed at the edges and clouds of the IoT [26]. Embedded microprocessors are the core part of video surveillance systems, used to manage image acquisition sensors to collect and preprocess image data.

4. Analysis of Experimental Results of DM Model of IoT on the Ground of Cloud Computing and EC.

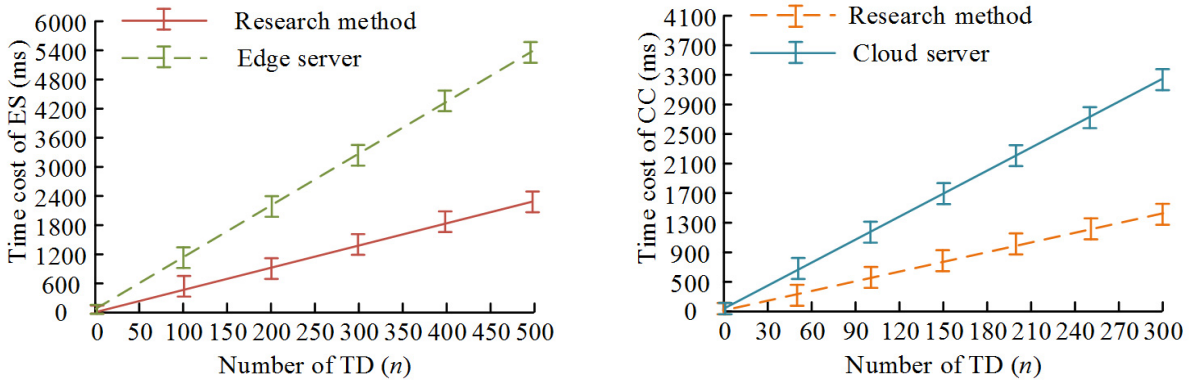
The hardware for experimental data collection included embedded microprocessors and image acquisition module sensors. Through the extensive utilization of CNNs such as AlexNet as well as ResNet in image classification tasks, it demonstrated the function of adding branch network structures. Test experiments were conducted to establish an NN using the open-source framework Chainer, using the Canadian Institute for Advanced Research-10 (CIFAR-10) dataset to simulate image data collected by IoT sensors. It divided the CIFAR-10 dataset into 5 subsets for training and 1 subset for testing, each containing 10000 images.

4.1. Experimental analysis of the effectiveness of collaborative methods on the ground of cloud computing and EC.

In the IoT edge DM model on the ground of EC model, the data collection module was designed using Raspberry Pi 3B+ as the IoT

TABLE 2. Hardware parameters of the Raspberry Pi 3B+

Hardware name	Argument
Soc	Broadcom BCM2837
CPU	ARM Cortex-A53 1.4GHz 64-bit quad-core
GPU	Broadcom VideoCrore lV
Internal memory	1GB (LPDDR2 400GHz)
Memory USB 2.0 port	4 × USB Ports
Video input	15-Needle MIPI Camera (CSI) interface
GPIO interface	40PIN
Network interface	10/100M Ethernet interface, 802.11n Wireless LAN
Bluetooth	Bluetooth 4.2
SD card interface	MicroSD card interface
Rated power	800 mah (W) 4.0
Power input	5V 2.5A/via MicroUSB or GPIO head



(a) Computing overhead for edge servers

(b) Computing overhead of cloud servers

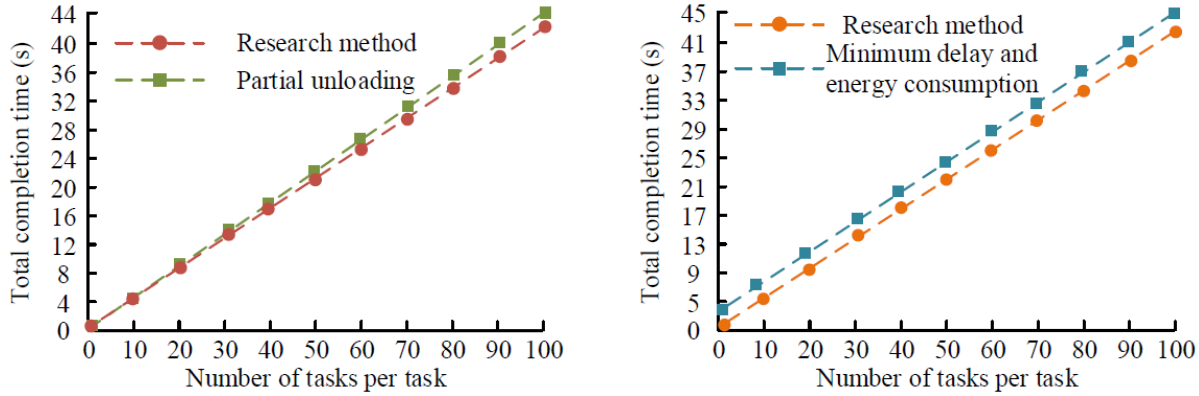
FIGURE 7. Computing overhead for each server

edge development platform. By connecting the image acquisition sensor to collect data, the specific parameters are showcased in Table 2.

For the convenience of scheme comparison, the number of edge servers ES in the system was set to t, and the number of terminal devices TD in each ES jurisdiction area was set to n. The length of timestamp ts was 64 bits, and the length of identity ID was 32 bits. The comparison of computing costs between edge servers and EC servers using the same research method is shown in Figure 7.

Figure 7(a) shows that as the TD of terminal devices increased, the time spent on research methods and individual edge servers also increased. However, the cost of research methods was increasing compared to individual edge servers, and when the number of TDs was 500, the difference reached its maximum value of approximately 3000 ms. Figure 7(b) shows that the cost of research methods was increasing compared to individual cloud servers. When the number of TD was 300, the difference reached its maximum value, approximately 1900 ms. In summary, the computing cost on the ground of cloud edge collaboration method was the most cost-effective. By setting different task volumes, the system latency and task completion time required for different offloading strategies under different offloading task volumes were compared. The result is shown in Figure 8.

Figure 8(a) shows that the total delay of task completion on the ground of the collaborative method of cloud computing and EC was far lower than the local unloading method



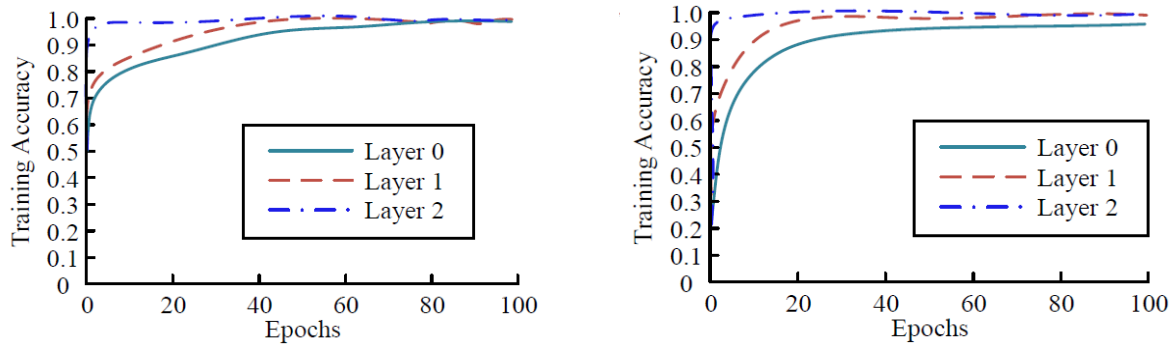
(a) Task completion time under different uninstallation policies

(b) Comparison of task completion time under different uninstall strategies

FIGURE 8. System time delay and task completion time under different uninstall tasks

under different number of unloading tasks. The research method reduced the latency by 4.1% compared to the local unloading method. This indicated that in actual deployment, the collaboration methods can effectively reduce task latency and improve service quality. Figure 8(b) shows that the research method always had lower latency than the minimum latency and energy consumption offloading strategy, and the latency was reduced by 9.9%. This indicated that in situations with high latency requirements, offloading strategies on the ground of research methods could better reduce latency.

4.2. **Performance analysis of IoT edge DM model on the ground of EC model.** The training accuracy of AlexNet and Resnet networks with branches is shown in Figure 9.

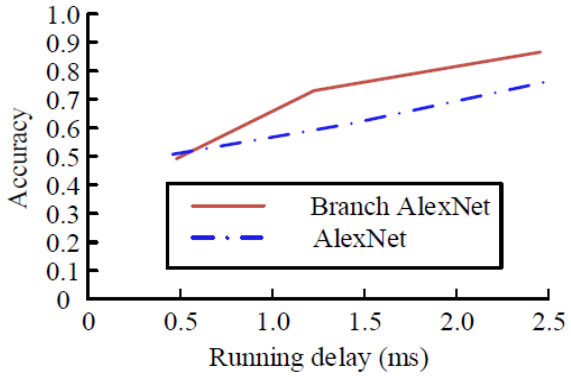


(a) Branch AlexNet network training accuracy

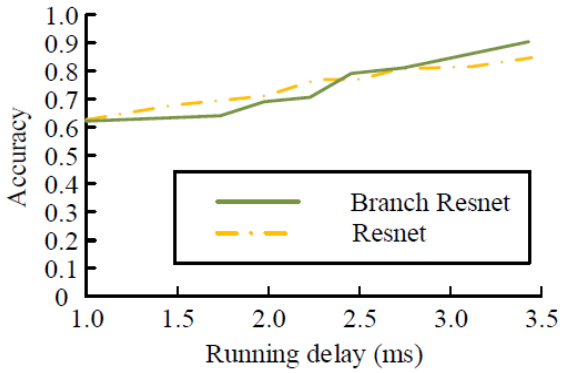
(b) Branch Resnet network training accuracy

FIGURE 9. Training accuracy of AlexNet and Resnet networks with branches

Figure 9(a) shows that for AlexNet networks with branches, the training accuracy of each layer also increased as the growth of training times. The training accuracy of Layer 2 was consistently higher than that of Layer 0 and Layer 1. And the training of Layer 2 first reached a stable state at 5 times, and after that, its training accuracy remained within the range of 0.98-1.0. Figure 9(b) shows that for training Resnet networks with branches, the training accuracy of Layer 2 was consistently higher than that of Layer 0 and Layer 1. The training of Layer 2 first reached a stable state at 3 times, and after that,



(a) Accuracy and runtime of branch AlexNet network classification

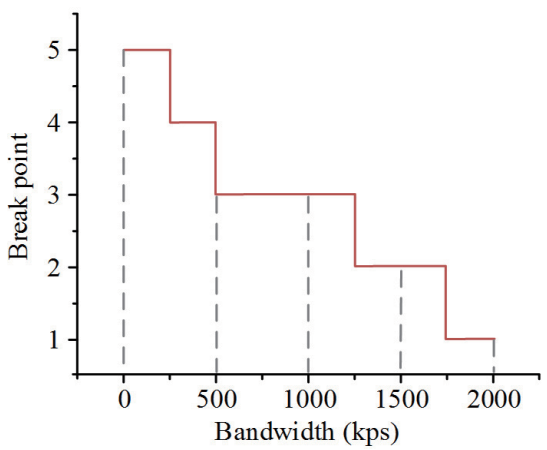


(b) Accuracy and runtime of branch Resnet network classification

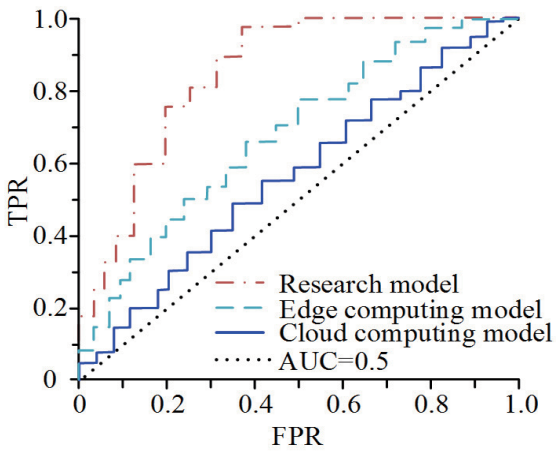
FIGURE 10. Accuracy and runtime of AlexNet and Resnet classification with branches

its training accuracy remained within the range of 0.99-1.0. The accuracy and runtime of AlexNet and Resnet network classification with branches are showcased in Figure 10.

Figure 10(a) shows that for AlexNet networks with branches, the classification accuracy of AlexNet networks with branches gradually exceeded that of AlexNet networks without branches after a running delay of 0.5 ms. When the latency was 1.25 ms, the classification accuracy of the branch AlexNet network differed the most from that of the unbranched AlexNet network by 15%. Figure 10(b) shows that for Resnet networks with branches, the classification accuracy of Resnet networks with branches gradually exceeded that of Resnet networks without branches after a running delay of 2.5 ms. When the delay was 3.5 ms, the classification accuracy of the branch Resnet network was 4.1% higher than that of the unbranched Resnet network. In conclusion, on the same accuracy needs, the reasoning time required by the DNN with BEP was markedly decreased relative to the original NN model. The test experiment deployed the AlexNet network with multiple BEP, the changes of partition points under different bandwidths. The classification effects of the research model, cloud computing model, and EC model on image DM are shown in Figure 11. Among them, the false positive rate (FPR) is the proportion of false predictions



(a) Optimal exit point partition point



(b) Classification performance of each model

FIGURE 11. Changes of division points and classification effects of each model

that are positive examples (target class). True positive rate (TPR) is the proportion of samples that are actually positive cases that are correctly predicted to be positive cases. The values of FPR and TPR together form the AUC curve, and the closer the value is to 1, the better the performance of the classifier.

Figure 11(a) shows that as bandwidth increased, the optimal exit point became lower. When the bandwidth was 0, the optimal point exit was 5. With sufficient bandwidth, more parts of the NN model would be divided into cloud servers for operation. When the bandwidth was 2000kps, the optimal point exit was 1. Without sufficient bandwidth, more parts of the NN model would be divided into edge devices for execution. Due to the growth of neural layers at the edge, the accuracy of the inference results was increased. Figure 11(b) showcases that the classification inference of the IoT edge DM model on the ground of EC model designed in the study was the best, with an AUC value of 0.92. The AUC values of cloud computing model and EC model were 0.68 and 0.73, respectively. This indicates that the research model has high performance in image DM and classification.

5. **Conclusion.** To alleviate the pressure of network bandwidth, reduce data processing latency, and solve the problem of DNN programs used for data classification processing being difficult to directly apply in IoT edge devices on EC platforms, this study constructed an IoT edge DM model on the ground of EC models. It utilized cloud edge collaborative processing for enhancing the efficiency and accuracy of data classification. The results showed that the cost of the research method was smaller than that of individual edge servers and cloud servers, with the maximum difference reaching 3000 ms and 1900 ms, respectively. The research method had a 4.1% reduction in latency compared to local offloading methods, and a 9.9% reduction in latency compared to the minimum latency and energy consumption offloading strategies. For AlexNet and Resnet networks with branches, the training accuracy of each layer also increased with the growth of training times. The training accuracy of Layer 2 was maintained within the range of 0.98-1.0 and 0.99-1.0, respectively. When the latency was 1.25 ms, the classification accuracy of the branch AlexNet network differed the most from that of the unbranched AlexNet network by 15%. When the delay was 3.5 ms, the classification accuracy of the branch Resnet network was 4.1% higher than that of the unbranched Resnet network. As bandwidth increased, the optimal exit point would become lower. The EC-based IoT edge DM model designed for research had the best classification and inference performance, with an AUC value of 0.92. In summary, the research method can markedly decrease the latency of computing systems, decrease data transmission bandwidth, and improve the efficiency of DM classification inference. However, the research on utilizing cloud edge collaborative processing of IoT data did not take into account device power consumption, so further improvement is needed in future research.

REFERENCES

[1] Y. Fang, B. Luo, T. Zhao, D. He, B. Jiang and Q. Liu, ST-SIGMA: Spatio-temporal semantics and interaction graph aggregation for multi-agent perception and trajectory forecasting, *CAAI Transactions on Intelligence Technology*, vol.7, no.4, pp.744-757, 2022.

[2] P. Sunhare, R. R. Chowdhary and M. K. Chattopadhyay, Internet of Things and data mining: An application oriented survey, *Journal of King Saud University – Computer and Information Sciences*, vol.34, no.6, pp.3569-3590, 2022.

[3] Y. Zhong, L. Chen, C. Dan and A. Rezaeipanah, A systematic survey of data mining and big data analysis in Internet of Things, *The Journal of Supercomputing*, vol.78, no.17, pp.18405-18453, 2022.

[4] M. A. Sulaiman, Evaluating data mining classification methods performance in Internet of Things applications, *Journal of Soft Computing and Data Mining*, vol.1, no.2, pp.11-25, 2020.

[5] C. Xie, X. Xiao and D. K. Hassan, Data mining and application of social e-commerce users based on big data of Internet of Things, *Journal of Intelligent and Fuzzy Systems*, vol.39, no.1, pp.1-11, 2020.
[6] M. T. Wu, S. Liu and C. W. Lin, Efficient uncertain sequence pattern mining based on Hadoop platform, *Journal of Circuits, Systems and Computers*, vol.31, no.15, pp.153-167, 2022.
[7] L. Li, Real time auxiliary data mining method for wireless communication mechanism optimization based on Internet of Things system, *Computer Communications*, vol.160, no.7, pp.333-341, 2020.
[8] M. Azrour, J. Mabrouki, A. Guezzaz and Y. Farhaoui, New enhanced authentication protocol for Internet of Things, *Big Data Mining and Analytics*, vol.4, no.1, pp.1-9, 2021.
[9] C. Chen, J. Wu, H. Lin, W. Chen and Z. Zheng, A secure and efficient blockchain-based data trading approach for Internet of Vehicles, *IEEE Transactions on Vehicular Technology*, vol.68, no.9, pp.9110-9121, 2019.
[10] F. Xue, Q. Hai, T. Dong, Z. Cui and Y. Gong, A deep reinforcement learning based hybrid algorithm for efficient resource scheduling in edge computing environment, *Information Sciences: An International Journal*, vol.608, pp.362-374, 2022.
[11] X. Bai, S. Chen, Y. Shi, C. Liang, X. Lv and F. R. Yu, Detection and defence method of low-rate DDoS attacks in vehicle edge computing network using information metrics, *International Journal of Sensor Networks*, vol.40, no.1, pp.20-33, 2022.
[12] X. Gao, R. Liu and A. Kaushik, Virtual network function placement in satellite edge computing with a potential game approach, *IEEE Transactions on Network and Service Management*, vol.19, no.2, pp.1243-1259, 2022.
[13] J. Yang, H. Li, J. Zou, S. Jiang, R. Li and X. Liu, Concrete crack segmentation based on UAV-enabled edge computing, *Neurocomputing*, vol.485, pp.233-241, 2022.
[14] X. Zheng, S. B. H. Shah, A. K. Bashir, R. Nawaz and U. Rana, Distributed hierarchical deep optimization for federated learning in mobile edge computing, *Computer Communications*, vol.194, no.10, pp.321-328, 2022.
[15] Y. Pang, J. Wu, L. Chen and M. Yao, Energy balancing for multiple devices with multiple tasks in mobile edge computing, *Journal of Frontiers of Computer Science and Technology*, vol.16, no.2, pp.480-488, 2022.
[16] B. Fang, M. Jiang, J. Shen and B. Stenger, Deep generative inpainting with comparative sample augmentation, *Journal of Computational and Cognitive Engineering*, vol.1, no.4, pp.174-180, 2022.
[17] Y. Tian, D. Su, S. Lauria and X. Liu, Recent advances on loss functions in deep learning for computer vision, *Neurocomputing*, vol.124, no.8, pp.1-14, 2022.
[18] W. L. Chin, Q. Zhang and T. Jiang, Low-complexity neuron for fixed-point artificial neural networks with ReLU activation function in energy-constrained wireless applications, *IET Communications*, vol.15, no.7, pp.917-923, 2021.
[19] G. Gous, P. D. Vaal and G. V. Coller, Implementation of averaging level control using native DCS functions, *Hydrocarbon Processing*, vol.100, no.5, pp.75-80, 2021.
[20] R. G. Brereton, Principal components analysis with several objects and variables, *Journal of Chemometrics*, vol.37, no.4, pp.1-9, 2023.
[21] F. Lu, L. Xu, Y. Jiang, P. Luo, X. Hu and J. Liang, Smear character recognition method of side-end power meter based on PCA image enhancement, *Nonlinear Engineering*, vol.11, no.1, pp.232-240, 2022.
[22] Z. C. Xin, J. S. Zhang, J. G. Zhang, J. Zheng, Y. Jin and Q. Liu, Predicting temperature of molten steel in LF-refining process using IF-ZCA-DNN model, *Metallurgical and Materials Transactions B*, vol.54, no.3, pp.1181-1194, 2023.
[23] S. S. Yeo, S.-R. Rho, H. J. Kim, J. Safdar, U. F. Zia and M. Y. Durrani, A triplet-branch convolutional neural network for part-based gait recognition, *Comput. Syst. Sci. Eng.*, vol.47, no.11, pp.2027-2047, 2023.
[24] X. Chu, H. Jiang, B. Li, D. Wang and W. Wang, Editorial: Advances in mobile, edge and cloud computing, *Mobile Networks & Applications*, vol.27, no.1, pp.219-221, 2022.
[25] S. E. Chafi, Y. Balboul, M. Fattah, S. Mazer and M. E. Bekkali, Enhancing resource allocation in edge and fog-cloud computing with genetic algorithm and particle swarm optimization, *Intelligent and Converged Networks*, vol.4, no.4, pp.273-279, 2023.
[26] A. Aldaej, IoT-inspired smart healthcare framework for diabetic patients: Fog computing initiative, *International Journal of Innovative Computing, Information and Control*, vol.18, no.3, pp.917-939, 2022.

Author Biography

Ning Hu with a bachelor's degree in Engineering, holds the position of associate professor, software engineer, and second-level psychological counselor. He serves as the dean of the School of Economics and Management at Kaifeng Vocational College of Culture and Arts. In 2007, he was honored as an outstanding educator in Kaifeng, and for three consecutive years, he was also awarded the title of leading young science and technology figure. He has been engaged in teaching since 1996, and has taught more than ten courses in computer information management and economics. His research direction is computer information management.