IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 November 2023, accepted 11 December 2023, date of publication 29 December 2023,
date of current version 8 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3348133

RESEARCH ARTICLE

IoT Fog Computing Optimization Method Based on Improved Convolutional Neural Network

BING JING AND HUIMIN XUE
Department of Internet of Things Technology, Shanxi Vocational and Technical College of Finance and Trade, Taiyuan 030031, China
Corresponding author: Huimin Xue (xhmlss2023@163.com)

ABSTRACT The rapid development of communication technology has promoted the development of the Internet of Things technology. It has resulted in a scarcity of computing resources for the Internet of Things devices, and limited the further development of the Internet of Things. In order to improve the utilization efficiency of the system resources for the Internet of Things devices and promote the further development of the Internet of Things, the continuous Markov decision process model is constructed. The value function approximation algorithm of the convolutional neural network is used to solve the problem. Continuous Markov decision process model is an excellent single-user decision process model, but not optimal for multi-user systems. Using convolutional neural network to solve the value function of continuous Markov decision process model, so that it can be applied to multi-user system. The results show that the average algorithm has growth rates of 0.48 and 0.84, respectively, in comparison to the other two algorithms. The average arrival rate has the least effect on the average delay of the value function approximation algorithm and the greatest influence on its power consumption. With the average arrival rate, the average delay of the algorithm increased by 0.25S and the power consumption by 0.27W. The effectiveness of the value function approximation algorithm based on convolutional neural network surpasses that of the multi-user multi-task offloading algorithm and the queue-aware algorithm, thus applying continuous Markov decision process models to multi-user systems. The study combines the continuous Markov decision process model with the resource decision of IOT devices, resulting in optimized resource scheduling decisions and improved utilization efficiency of IOT devices.

INDEX TERMS MDP, CNN, value function, IoT, fog computing.

I. INTRODUCTION

With the continuous development of communication technology, many devices are becoming more and more intelligent. Sensor networks are embedded in smart devices. Smart devices can collect environmental information in real time with the help of sensor networks, and upload these information to the Internet to realize the connection and interaction of smart devices. Based on this phenomenon, Auto-ID proposed the concept of ''The Internet of Things (IOT)'' in 1999. IOT can connect people and things through network access. As a result, smart devices can collaborate via the internet to accomplish various functions [1], [2]. However, due to the explosive growth of IOT, the traditional cloud computing data processing methods cannot meet the data processing requirements of IOT. To solve this problem, scholars first proposed the concept of ''fog computing'' based on ''cloud computing'' in 2012, that is, adding the network edge layer between smart devices and cloud data processing center. This approach significantly enhances the transmission efficiency and reduces the latency [3], [4]. The emergence of atomized computing systems has greatly improved the data processing capacity of the IOT. However, it has also resulted in a depletion of computing and storage resources for the IOT devices. Therefore, this data processing method needs to deal with offloading and unlimited allocation of resources. MDP is a commonly used method to obtain the best control strategy in random task reach models, but it is extremely difficult to apply in multi-user systems. In order to make the decision maker can be applied to multi-user system, a convolutional

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

neural network (CNN) based on value function approximation algorithm (VFAA) is proposed. This VFAA solves the application problem of MDP in multi-user systems, thereby enabling its use in such scenarios.

The primary achievement of this research lies in its pioneering efforts to broaden the application of the MDP specifically tailored for multi-user systems. By doing so, it has opened new avenues for system optimizations. Furthermore, the study introduces innovative methodologies that harness the potential of IOT devices. It results in significant enhancements in the efficiency of allocated system resources. These advancements not only serve the theoretical realm but also have practical implications in enhancing multi-user system operations. The paper innovatively proposes the application of continuous Markov decision process and multi-user system, and the use of CNN based VFAA to solve it.

The study will be conducted from four parts. Section II provides an overview of IOT-related research. Section III presents an analysis of an optimized fog computing technique for IOT using improved CNN. Section IV details the experimental verification of this method, and Section V summarizes the research conducted.

II. RELATED WORKS

As the growth of IOT, the traditional cloud computing data processing method cannot satisfy the computing demands of IOT, based on this, Lakhan et al. proposed a federated learning fog computing model based on digital twin technology to enable hospital equipment to connect to the IOT, improve the efficiency of healthcare resource utilization, and reduce task processing time and failure risk. The author also proposed a fog computing framework that considers security and fault tolerance strategies. The results demonstrate a 40% reduction in security risks and a 50% reduction in task failure risks with the proposed fog computing model [5]. Mohammed et al. proposed a cancer detection method based on fog heterogeneous computing nodes to improve the detection efficiency of cancer by utilizing distributed fog computing. They also proposed a cancer multi omics spiritual bed dataset for model learning. The results show that the detection accuracy of the method proposed by the author is 98%, and the processing delay is 61% [6]. Mohammed et al. proposed a distributed blockchain network task offloading and scheduling system to improve data processing capabilities and security in medical systems. The results showed that the method proposed by the author reduced the data processing power consumption of existing medical systems. It results in a 39% reduction in training and testing time and improved data security [7]. Lakhan et al. proposed a fuzzy energy-saving decision support system to improve the efficiency of energy and resource utilization in transportation applications. The results show that the method proposed by the author can effectively reduce energy consumption, task latency, and transmission costs. It enhances the scheduling accuracy of the application [8]. Mutlag et al. proposed a fog computing resource management model to optimize task scheduling in existing healthcare systems in order to solve the scheduling problem of task resources by devices in healthcare systems. The results show that the model proposed by the author outperforms existing methods in terms of task latency, response or component, transmission cost, and power consumption [9].

CNN is often combined with various types of functions in applications to improve their potential. Chen et al. proposed a generalized frequency response function combined with CNN to improve the accuracy of traditional fault diagnosis methods. The results showed that the new method had an accuracy of 0.9875, thus confirming the reliability of conventional fault diagnosis methods [10]. Pezzano et al. put forward a CNN segmentation strategy combined with loss function to optimize the segmentation problem of lung nodules in computed tomography images. The results showed that the new method has an F1 score and IoU aspects of 0.033 and 0.047, respectively. They are significantly better than the existing CT nodule segmentation techniques and also increases the detail in nodule boundaries, even under the noisiest conditions [11]. Xiong et al. put forward a four-layer CNN based on loss function to preserve the spatial and spectral information of the original panchromatic, and multispectral images. The outcomes showed that the method attained superior performance in both subjective visual effects and objective evaluation [12]. Sharma et al. to make image enhancement and adjustment techniques that can improve the quality and characteristics of images proposed an edge adjustment CNN with S-shaped functions. The results prove that the method has better performance. However, due to the limited amount of information contained in a single image, the method cannot work efficiently [13]. In order to improve the recognition rate of finger vein, Xie et al. proposed a new algorithm for finger vein recognition using CNN and supervised discrete hash algorithm. The results show that this method significantly enhances the precision of recognition when compared to alternative finger vein recognition methods [14].

In summary, the optimization of the offloading and infinite resource allocation problems in IOT fog computing remains an ongoing challenge. Markov Decision Process (MDP) has high applicability in such problems but cannot meet the application in multi-user systems. The CNN combined with functions can help MDP solve the problem that cannot be applied to multi-user systems. Therefore, the article uses CNN-based VFAA to solve and analyze MDP. Consequently, it can be applied in multi system to solve the offloading and infinite resource allocation of IOT fog computing.

III. ANALYSIS OF IOT FOG COMPUTING OPTIMIZATION METHOD BASED ON IMPROVED CNN

A. MDP-BASED IOT FOG COMPUTING OPTIMIZATION MODEL CONSTRUCTION

In IOT, the comprehensive resources of the device are limited, but the computing volume of the device is increasing. This leads to the contradiction between the number of computing resources and the computing resources of the device in

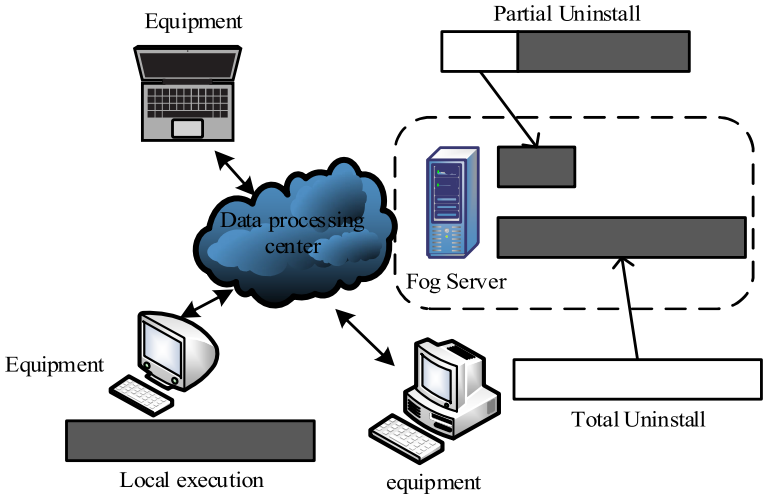

FIGURE 1. Results of centralized offloading decisions for IOT devices.

IOT. However, this issue can be resolved through computation offloading technology. In IOT, there are more factors affecting computational task offloading. The Communication overhead and computation overhead are the performance indicators of computational task offloading. They reflect the operational performance of the application. The computational task offloading process of IOT devices comprises four steps: task submission, task remote execution, offloading decision result and computation result feedback, where the offloading decision result is shown in Figure 1 [15].

Since partial offloading involves performing part of the computation locally and offloading the rest to a fog computing server for processing. It is a complex process influenced by various factors, such as the type of application and the data to be offloaded. However, the article focuses solely on full offloading and local offloading when optimizing IOT fog computing. The IOT device includes sensing, computing and communication as the main modules. The data collected during its operation needs to be processed to produce useful results that can be used to control the device and execute commands. However, due to the extremely limited computing power of the device, the data it collects needs to be offloaded to the fog computing server for processing. Figure 2 depicts the IOT fog computing system model.

The IOT fog computing system model comprises communication model and computation model. The communication model is responsible for wireless resource control, including scheduling, link adaption and power control. The computation model is divided into the device computation model and fog server computation model. The article mainly considers the multi-user fog computing system for IOT. To simplify the MDP model, it formulates a continuous-time MDP model for the IOT, in which each packet records the global system state (GSS) when it moves. During decision moments of the MDP model, it is necessary to select the behavior for the system state at that moment. When the packet moves, it will cause the system to move from the current state to another state. The article defines the GSS as S_k and the behavior of the k decision period (DP) as $a_k = (a_{o,k}, a_{s,k})$, where $a_{o,k} \in A_0 = \{-1, 0, 1\}$ denotes the offloading behavior and $a_{s,k} \in A_S = \{0\} \cup N$ denotes scheduling behavior, then the state-dependent unloading behavior is as in Equation (1) [16].

$$A_{o,s_k} = \begin{cases} \{0\}, & n_k = -N \cup \{0\}, or \\ & n_k = n \in N, Q_{n,k} = M, Q_{n,k}^{loc} < M^{loc} \\ \{1\}, & n_k = n \in N, Q_{n,k} < M, Q_{n,k}^{loc} = M^{loc} \\ \{-1\}, & n_k = n \in N, Q_{n,k} = M, Q_{n,k}^{loc} = M^{loc} \\ \{0, 1\}, & otherwise \end{cases} \tag{1}$$

In Equation (1) A_{o,s_k} denotes the state-dependent offload behavior, n_k denotes the state, $Q_{n,k}$ denotes the transmission queue (TQ) length vector for offloading to the fog server appeared at the very start of the k DP when the packet movement first occurs, M denotes its maximum length, $Q_{n,k}^{loc}$ denotes the processing queue (PQ) length vector for local computation, M^{loc} denotes its maximum length, and n denotes the IOT device, satisfying $n \in N = \{1, 2, 3, \ldots, N\}$. Since the packet arrival event occurs in IOT devices, according to the offloading behavior, the arriving packets need to be added to the TQ or PQ of the device n. They will result in the length of the TQ and PQ after the offloading behavior of at the $S_k k$ DP with the values of $Q_{n,k}$, $Q_{n,k}^{loc}$, at this time, the TQ length of the device n after the offloading decision at the k DP is Equation (2).

$$\tilde{Q}_{n,k}^{loc} = \begin{cases} Q_{n,k}^{loc} + 1, & \ell_k = n \in N, a_{o,k} = 0 \\ Q_{n,k}^{loc}, & otherwise \end{cases} \tag{2}$$

In Equation (2), $\tilde{Q}_{n,k}^{loc}$ represents the TQ length of the device after the offload decision in the DP. The PQ length of the device n after the offload decision at the k DP is Equation (3).

$$\tilde{Q}_{n,k} = \begin{cases} Q_{n,k} + 1, & \ell_k = n \in N, a_{o,k} = 1 \\ Q_{n,k}, & otherwise \end{cases} \tag{3}$$

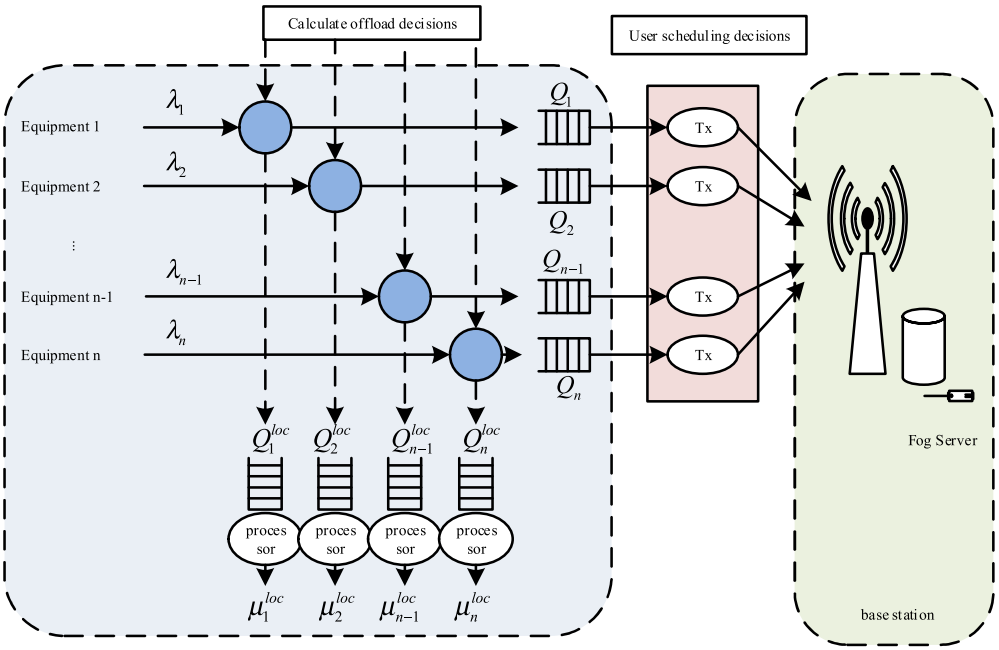

FIGURE 2. A Schematic diagram of the IOT fog computing system model.

In Equation (3), $\tilde{Q}_{n,k}$ represents the PQ length after the offload decision of the device at the nk DP. At this point, the post-decision transfer queue and PQ for the DP is $k\tilde{Q}_k = \left\{\tilde{Q}_{n,k}\right\}_{n=1}^{N}, \tilde{Q}_k^{loc} = \left\{\tilde{Q}_{n,k}^{loc}\right\}_{n=1}^{N}$. Based on the above values, $a_{s,k}$ is defined such that if $a_{s,k} = 0$, then it denotes that no queue is planed and if $a_{s,k} = n \in N$, then it means that the TQ n is scheduled and the scheduling behavior is considered only when packets leave from the TQ or packets arrive and no queue is scheduled. When the above happens, the device to be scheduled is selected from the IOT devices with non-empty TQ, or else, it is similar to the previous DP b_k. So the set of state-dependent scheduling behaviors is shown in Equation (4).

$$A_{s,S_k} = \begin{cases} N_{s,k}, & n_k = 0 or\ (n_k \in N\ and\ b_k = 0) \\ \{b_k\}, & otherwise \end{cases} \tag{4}$$

In Equation (4) A_{s,S_k} denotes the state-dependent scheduling behavior. The article defines the post-decision GSS of the k th DP as $\tilde{S}_k$, then the deterministic function of S_k and the behavior of for the $a_k k$ th DP is Equation(5).

$$\tilde{S}_k = f\left(s_k, a_k\right) = \left(\tilde{Q}_K, \tilde{Q}_k^{loc}, \ell_k, a_{s,k}\right) \tag{5}$$

The post-decision GSS is the state after the optimal policy has been determined, and its state space is the same as that of the GSS. The transfer from the decision state to the post-decision state of the continuous MDP is shown in Figure 3 [3].

The duration of the k DP is equivalent to the sojourn time of the continuous-time MDP model at S_k given the behavior a_k, which obeys an exponential distribution with parameter $\beta\left(s_k, a_k\right)$ and $\beta\left(s_k, a_k\right)$ as shown in Equation (6).

$$\beta\left(s_k, a_k\right) = \sum_{n=1}^{N} \lambda_n + \mu_{a_{s,k}} + \sum_{n=1}^{N} \mu_n^{loc} 1_{\left(\tilde{Q}_{n,k}^{loc} \neq 0\right)} \tag{6}$$

In Equation (6) λ_n is the average packet arrival rate. $\mu_{a_{s,k}}$ denotes the fog computing system data processing time. μ_n^{loc} denotes the local data processing time, which can also be represented by the post-decision state. To retrace the payoff function of the continuous MDP model, it is necessary to observe the strategy that shortens the weighted sum (WS) of the average delay (AD) and power consumption (PC) of all IOT devices Ω. Ω is a function by which the decision maker can determine the behavior at the state $s_k a_k = \Omega\left(s_k\right) \in A_{o,s_k} \cup A_{s,s_k}$, whose associated behavior space has been given in Equation (1), Equation (4). And this study considers the above dynamic optimization problem as a wireless average payoff continuous MDP problem with the optimization objective of Equation (7).

$$\begin{aligned} \min_{\Omega} J\left(\Omega\right) &= \sum_{n=1}^{N} \left(\omega_n \bar{D}_n + \gamma_n \bar{P}_n\right) \\ &= \lim_{T \to \infty} \frac{E^{\Omega}\left[\sum_{k=0}^{T} \int_{\sigma_k}^{\sigma_{k+1}} c\left(s_k, \Omega\left(s_k\right)\right) dt\right]}{E^{\Omega}\left[\sum_{k=0}^{T} \tau_k\right]} \end{aligned} \tag{7}$$

In Equation (7) $\omega_n \min_{\Omega} J\left(\Omega\right)$ denotes the optimal objective. ω_n, γ_n are the weights of the AD. $\bar{D}_n$ and PC $\bar{P}_n$ of the device n respectively. $E^{\Omega}[.]$ denotes the function Ω taking the expectation, σ_k, τ_k are the begin point and length of time of the k DP respectively. $c\left(s_k, \Omega\left(s_k\right)\right)$ denotes when the system is at S_k and the cost is appeared at this rate by choosing the behavior $\Omega\left(s_k\right)$ in the k DP. To complete the expression of

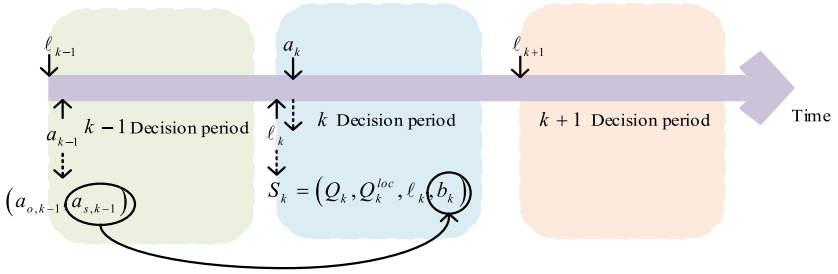

FIGURE 3. Schematic of state transition of the continuous MDP model.

Equation (7), it is necessary to obtain $c\left(s_k, \Omega\left(s_k\right)\right)$, the local processing delay of the packet is Equation (8).

$$\bar{D}_n^{loc} = \frac{E^{\pi(\Omega)}\left[\tilde{Q}_{n,k}^{loc}\right]}{\lambda_n p_{n,noff}} \tag{8}$$

In Equation (8) $p_{n,noff}$ is the local processing probability of the device n and $E^{\pi(\Omega)}[.]$ denotes the expectation taken for the policy Ω. The remote processing of packets is divided into uplink transmission, remote calculation and downlink transmission delay. Therefore, the AD of remote processing of the device n be derived as Equation (9).

$$\bar{D}_n^{rem} = \frac{E^{\pi(\Omega)}\left[\tilde{Q}_{n,k}\right]}{\lambda_n p_{n,off}} \tag{9}$$

In Equation (9), $p_{n,off}$ is the offload probability of the device n. According to Equation (8) and Equation (9), the AD of the device n can be inferred as Equation (10). The average PC of IOT devices can be obtained by the same method of derivation, the article does not go into too much detail.

$$\bar{D}_n = E^{\pi(\Omega)}\left[\frac{\tilde{Q}_n + \tilde{Q}_n^{loc}}{\lambda_n}\right] \tag{10}$$

Combining the average latency and PC of IOT devices and their weights, the expression $c\left(s_k, \Omega\left(s_k\right)\right)$ can be obtained. It is demonstrated in Equation (11).

$$c\left(s_k, \Omega\left(s_k\right)\right) = \sum\left(\begin{array}{l}\frac{\omega_n'}{\sum_{i=1}^{N}\lambda_l}\left(\tilde{Q}_{n,k} + \tilde{Q}_{n,k}^{loc}\right) + \\ \frac{\gamma_n'}{N}\left(P_n * 1_{a_{s,k}\neq n} + P_n^{loc} * 1_{\tilde{Q}_{n,k}^{loc}\neq 0}\right)\end{array}\right) \tag{11}$$

In Equation (11), $1_{condition}$ is a random variable. The continuous-time MDP theory and the Bellman equation can be utilized to calculate the posterior decision state value function of the optimal strategy for the continuous-time MDP model $V\left(\tilde{s}_k\right)$.

B. CNN-BASED VFAA

CNN is a common deep learning method in artificial neural networks. The article will use CNN-based VFAA to solve the value function of MDP model $V\left(\tilde{s}_k\right)$. Its approximation architecture is Equation (12).

$$V\left(\tilde{s}^{(i)}\right) \cong \sum_{n=1}^{N}\sum_{j=1}^{D}\phi_{\tilde{S}_n^{(j)}}\left(\tilde{S}^{(i)}\right)V_n\left(\tilde{S}_n^{(j)}\right) \tag{12}$$

In Equation (12), $\phi_{\tilde{S}_n^{(j)}}$ is the feature vector of the post-decision GSS $\tilde{S}_n^{(i)}$ and $V_n\left(\tilde{S}_n^{(j)}\right)$ is the single-node value function of the post-decision local system state (LSS) $\tilde{S}_n^{(j)}$. The CNN-based value function approximation architecture is shown in Figure 4 [17].

$\tilde{s}^{(i)}$ $(N * D)$ $((n-1)D + j)$ $j\tilde{s}^{(i)}N$ the quantity of neurons in the fully connected layer (FCL) is similar to those in the input layer (IL), and the corresponding state of each neuron is also the same as the IL. The output layer (OL) is used to infer the global value function (GVF) of $\tilde{s}^{(i)}$ based on the approximate architecture of the post-decision GVF. Let $c\left(\tilde{s}^{(i)}\right)$ be the N dimensional activation vector of the convolutional layer (CL), then the f weight $w^f_{m,\tilde{s}_n^j}$ is the link weight of the FCL $\tilde{s}_n^{(j)}$ from the first m neuron in the CL, then the activation vector is Equation (13) [18].

$$\bar{\phi}\left(\tilde{s}^{(i)}\right) = \sigma\left(c\left(\tilde{s}^{(i)}\right) * W^f\right) \tag{13}$$

In Equation (13), $\bar{\phi}\left(\tilde{s}^{(i)}\right)$ is the FCL activation vector. σ is the Tanh function, and is the W^f $(N * (N * D))$ dimensional matrix. The activation of the $((n-1)D + j)$ the neuron in the multiplication layer (ML) is the eigenvalue of the approximate architecture of the post-decision GVF $\phi_{\tilde{s}_n^{(j)}}$. $\phi\left(\tilde{s}^{(i)}\right)$ is dimensional, and the $N * D((n-1)D + j)$ the element is $\phi_{\tilde{s}_n^{(j)}}, \phi\left(\tilde{s}^{(i)}\right)$ which is expressed as Equation (14).

$$\phi\left(\tilde{s}^{(i)}\right) = \bar{\phi}\left(\tilde{s}^{(i)}\right)ex\left(\tilde{s}^{(i)}\right) \tag{14}$$

In Equation (14), e denotes the product of two vector elements and $x\left(\tilde{s}^{(i)}\right)$ presents the matrix of $\tilde{s}^{(i)}$. To explain and illustrate the CNN-based function approximation architecture, the article assumes that the post-decision GVF vector V, the f weight matrix W^f and the c weight vector w^c are obtained for each device. Then at the first k DP, the optimal behavior $a_k^* = \Omega^*\left(S_k\right)$ can be derived from the above obtained value functions and weights under the current GSS S_k according to Equation (15).

$\Omega^*\left(S_k\right)$

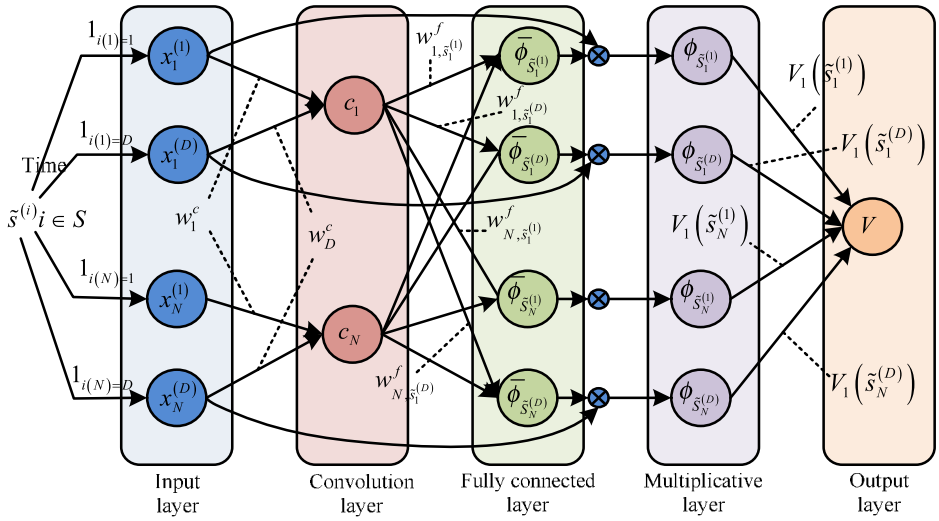

FIGURE 4. Function approximation architecture of CNN based VFAA.

$$= \arg\min_{\Omega} \sum_{n=1}^{N} \left(\tilde{g}_n \left(S_{n,k}, \Omega \left(S_k \right) \right) + \phi_{\tilde{s}_{n,k}} \left(\tilde{S}_k \right) V_n \left(f_n \left(S_{n,k}, \Omega \left(S_k \right) \right) \right) - \theta_{n,k} / \beta \left(S_k, a_k \right) \right) \quad (15)$$

In Equation (15), $\tilde{g}_n \left(S_{n,k}, \Omega \left(S_k \right) \right)$ is the return function of the continous MDP model. In designing the CNN architecture, the article first, obtains the approximate GVF by summing the local value function of each of all users and incorporates local features to rely on the global system. It can enhance the approximation accuracy. Second, the effectiveness of the learning algorithm can be improved by a CL to compress the LSS of each user into a single scalar. Third, the computational complexity and signaling overhead associated with parameter updates can be reduced by inserting a ML before the OL. Finally, an auction mechanism is used to bid on each IOT device to bidding according to its local behavior and decide on the optimal behavior. In the above approach, the global optimum is determined through a semi-distributed implementation that involves the base station architecture and IOT devices. The determination of the global optimum is illustrated in Figure 5 [19].

In the overall flow of decision operations and signaling overhead, each IOT device involves only one addition operation and one multiplication operation. While the base station must perform N addition operations, and the signaling overhead and base stations grow linearly with IOT devices. The overall solution's operational flow is depicted in Figure 6 [20]. In Figure 6, at the beginning of the scheme, chaotic variables are used to initialize the data, which can effectively avoid concentration during data initialization [21]. The data initialization, letting $k = 0$, indicates the DP index, initializing the single node value function vector. When the packet movement occurs, it is possible that the device of the event in the DP, the device of the event and the device of the newly scheduled packet in the DP need to update the LSS. While the base station only needs to notify the second and third devices for the LSS update. There are also three possibilities for the event to occur in the DP, i.e., determining the offload behavior, determining the scheduling behavior, and no need to determine the behavior. When the packet leaves the PQ of the device, it is not necessary to determine the behavior. All other possibilities must determine their behavior. After the optimal control behavior is confirmed, the post-decision system state update as well as the single node value function update for the device is required.

The training process of CNN is as follows: first, prepare the dataset and divide it into training, validation, and test sets. Secondly, preprocess the data, such as normalization and enhancement. Thirdly, build a model. Design a network structure consisting of convolutional layer, pooling layer, and fully connected layer, and initialize all parameters. During the model training phase, data is transmitted through the network, generating feature maps layer by layer until the prediction results are output. Use loss functions, such as cross-entropy to compare predictions to real labels and compute errors. Fourthly, initiate back-propagation. This process calculates the gradient of losses and updates weights based on optimizer such as Adam. This process is iterated numerous times throughout the dataset until the loss reaches convergence or the specified number of cycles.

IV. SIMULATION RESULTS ANALYSIS OF CNN-BASED VFAA

The optimal decision of the continuous MDP model is a classification problem. The Image Net dataset is a commonly used image recognition dataset. To verify the superiority of CNN in classification problems, a study was conducted on the Image Net dataset. The accuracy of image classification was compared between CNN and Support Vector Machine (SVM), Decision Tree Theory, Naive Bayes (NB), and k-Nearest Neighbor (KNN), as shown in Figure 7.

In Figure 7, as the number of samples increases, the correct rate of both the classification method and the decision tree

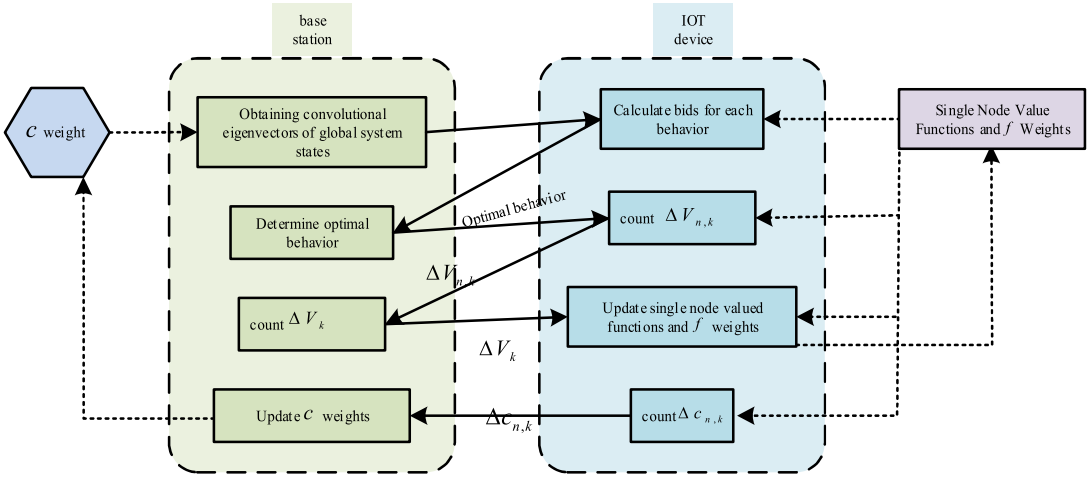

FIGURE 5. Semi-distribution implementation of CNN based VFAA.

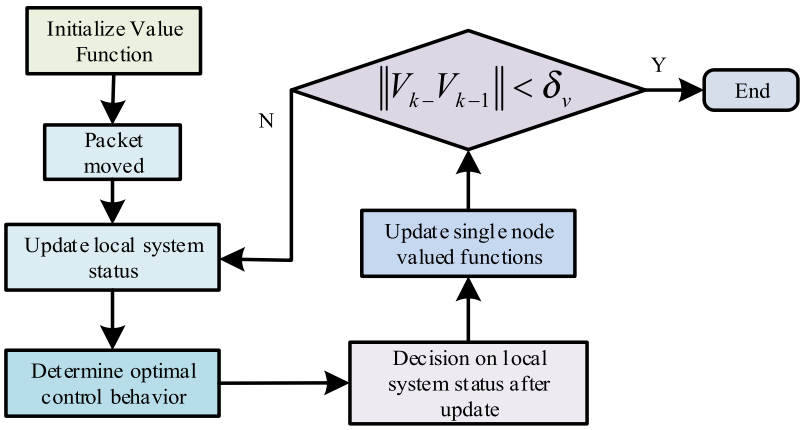

FIGURE 6. Overall solution implementation process.

classification method improves, except for the decision tree classification method. It also improves when the number of samples is less than 12. At a sample size of 20, the correct rate of CNN is the highest at 0.988 and the lowest at 0.821. When the sample size is 18, the maximum correct rate of CNN is 99.8. The maximum correct rate of support vector machine is 96.2 when the sample size is 16. The maximum correct rate of Bayesian classification is 0.953 when the sample size is 10. The maximum correct rates of both K-nearest neighbor and decision tree are 0.948 and 0.953 respectively when the sample size is 12. In conclusion, the CNN algorithm achieves higher accuracy than the other algorithms, and it is more suitable for the decision judgment of the IOT. Prior to simulating the CNN-based VFAA, the paper reviews cross-node communication latency and CPU utilization of mainstream CNN models. The results are shown in Figure 8.

In Figure 8, LeNet has the highest CPU utilization at 0.60, trailed by R-CNN and ZFNet with 0.55, and the lowest CPU utilization is U-Net with only 0.01, followed by FCN with 0.05. The highest cross-node communication latency is U-Net at0.40, followed by YOLO v3, MobileNet v2 and Incepeion v4 at0.23. The lowest cross-node communication latency is R-CNN with only 0.09, followed by LeNet with 0.095. To verify the superiority of the algorithms proposed in this paper, a discrete event system-level simulator for the NB IOT fog computing system was built using Matlab software on a Windows 7 system. A comparative experiment was conducted on the performance gap between the UCI Machine Learning Repository dataset and other traditional IOT fog computing algorithms. This dataset comprises numerous IOT-related data that can facilitate a broad array of IOT-related learning tasks. The study sets the transmission radius of the simulated IOT to 500m, divides the IOT area into 10. And it sets the repetition number of IOT devices in a single area to 1, sets the Path loss channel model to $15.3 + 37.6log_{10}l_k[m]$, l_k to $(d_{k-1} + d_k)/2m$, P_{CMAX} to 23dBm, X to 10^5, f_n^{loc} to $[1, 3] \times 10^9 cycles/\sec$, and L to 10kbits. The comparison results of the average latency, power consumption, and weighted sum of the two algorithms with the number of devices are shown in Figure 9.

In sub-figure (a), the CNN-based VFAA has the slowest growth rate of latency, with an increase in latency of only 0.65 seconds when the number of devices reaches 35. When the number of devices is from 0 to 10, the gap between the latency of the three algorithms is not large. They all grow slowly from 0.05S to 0.2S. When the number of devices

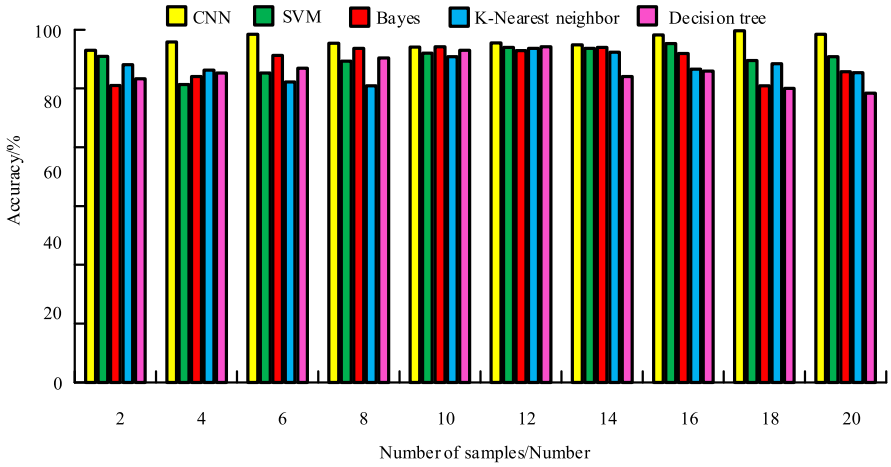

FIGURE 7. Comparison of accuracy rates of classification algorithms.

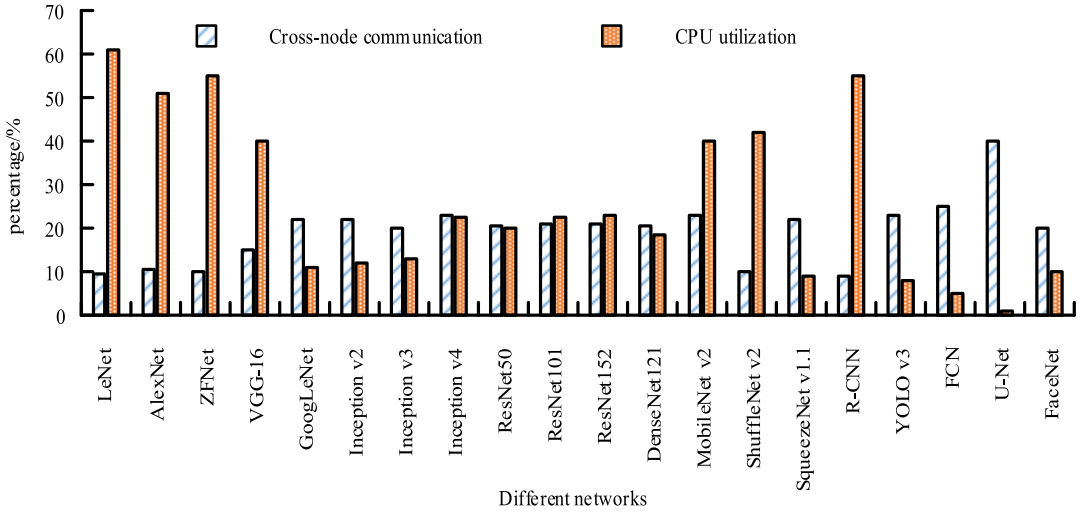

FIGURE 8. Cross node communication latency and CPU utilization of different networks.

exceeds10, the growth rate of the algorithm latency starts to open up the gap. As can be seen in sub-figure (b), the PC of the queue-aware algorithm grows the slowest. The PC only increases by 0.22W when the number of devices increases to 35. At 15 devices, the PC of all three algorithms is equivalent, measuring 0.09W. When the number of devices increases from 25 to 30, the PC of the CNN-based VFAA decreases from 0.27 to 0.265. The PC of all other algorithms increases. In sub-figure (c),the WS of AD and PC of the three algorithms are basically the same when the number of devices is less than 10. However, the difference between the algorithms becomes more significant when the number of devices exceeds 10. The WS of the AD and PC of the CNN-based VFAA is the lowest when the number of devices is 35, which is about 0.5. In conclusion, the proposed algorithm exhibits the lowest average delay and the lowest power growth. The AD, PC and WS of the two algorithms are compared with the average arrival rate of the three algorithms as shown in Figure 10.

In sub-figure (a), the difference in the AD growth among the four algorithms increases when the average arrival rate exceeds 1. The multi-user multi-service offload algorithm has the largest growth, and its AD increases by 0.6S when the average arrival rate increases from 1 to 2. The remaining algorithms have a lower increase, with the CNN-based VFAA having the smallest increase of 0.21S. In sub-figure (b), when the average arrival rate is less than 1, the PC of the linear VFAA and the CNN-based VFAA are the same. Both are lower than the other two algorithms. Among the four algorithms, the PC of the multi-user multi-service offloading algorithm increases more slowly at about 0.1. The remaining three algorithms increase more rapidly, with the queue-aware algorithm having the most significant increase at around 0.3. In sub-figure (c), the WS of the AD and PC with the average arrival rate of the three algorithms is basically the same trend. The growth of the CNN-based VFAA is the lowest, increasing only by about 0.25. In conclusion, the power consumption of the proposed algorithm is the lowest. Figure 11 displays the comparison results of AD, PC, and WS for the three algorithms while changing the delay weights.

In sub-figure (a), the AD of the queue-aware algorithm basically does not change with the change of the delay weight. When the delay weight is 0.25, the AD of the CNN-based

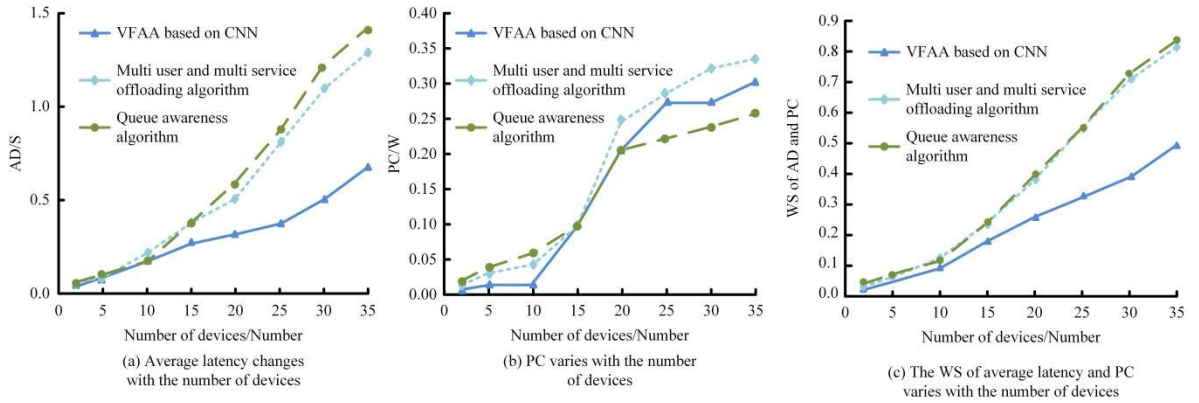

FIGURE 9. Comparison results of AD, PC and WS of the three algorithms under the number of devices.

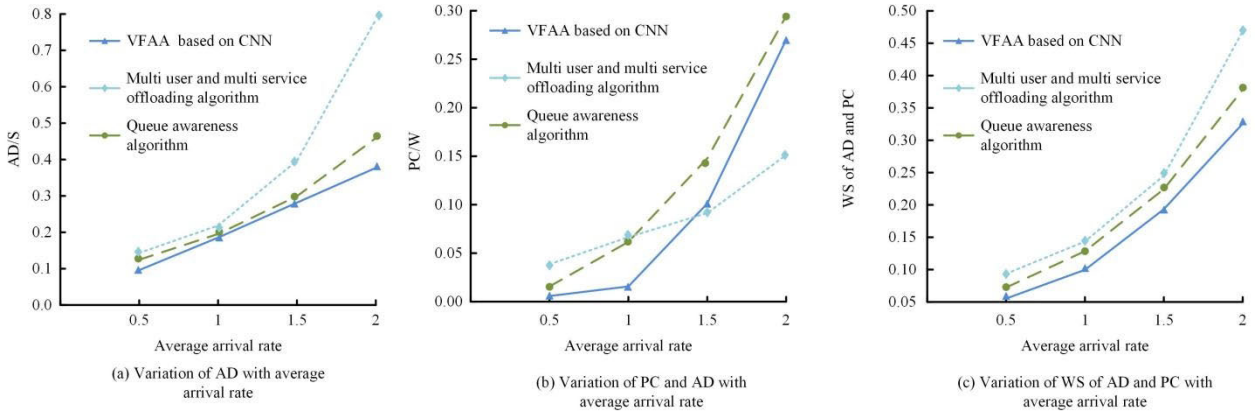

FIGURE 10. Comparison results of AD, PC and WS of the three algorithms under average arrival rate.

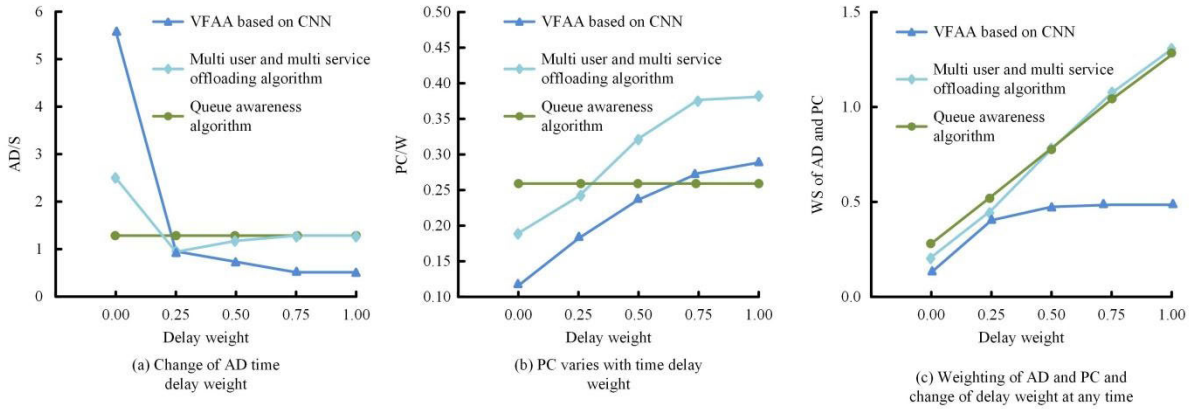

FIGURE 11. Comparison results of AD, PC and WS of the three algorithms under delay weighting.

VFAA and the multi-user multi-service offloading algorithm are consistent at about 1S. And then, with the increase of the delay weight, the AD of the CNN-based VFAA continues to decrease, while the AD of multi user and multi service offloading algorithms begins to rise. It can be seen in sub-figure (b) that the PC of the queue-aware algorithm also does not change with the delay weights. The PC of the algorithm increases with the delay weights. Additionally, it shows that the CNN-based VFAA has the lowest PC, with its highest value being around 0.27 W. It can be seen in sub-figure (c) that the delay weights of the CNN-based VFAA are from 0 to 0.25,while the WS of the AD and PC increase faster from 0.25 to 1, and it is basically unchanged from 0.5 to about 1. In conclusion, with the increasing delay weight, the proposed algorithm has the lowest power growth and the highest average arrival rate. The AD and WS of the algorithm increases linearly with the delay weight, and the maximum value is about 1.3. The article also conducts the convergence

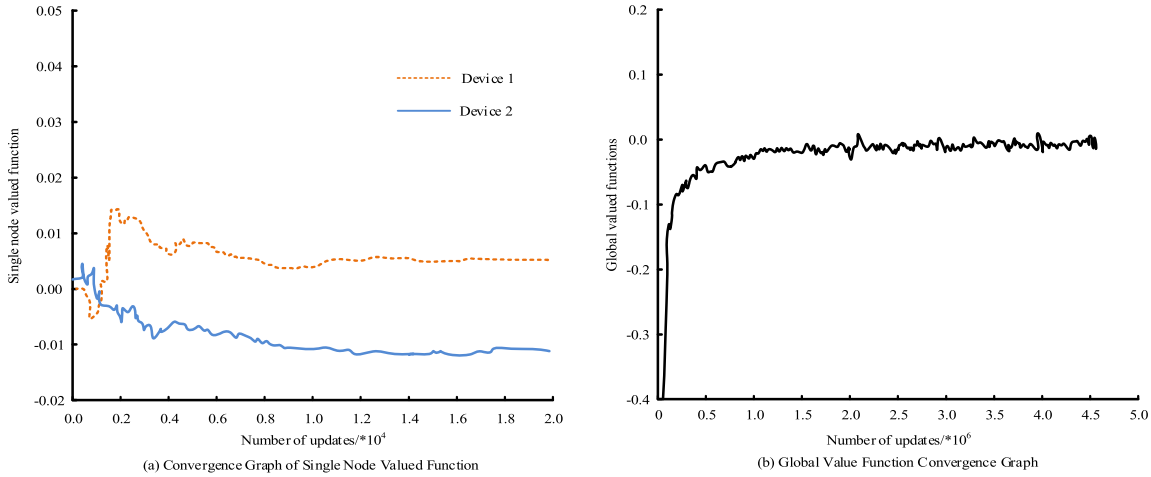

FIGURE 12. Convergence characteristics of CNN based VFAA.

TABLE 1. Comparison of task scheduling methods.

Algorithm	RMSE	Accuracy	Time（s）
CNN	0.1538	95.6%	0.68
Genetic algorithm	0.2167	91.5%	1.34
Greedy Algorithm	0.1867	92.6%	0.97

simulation experiments for the value function of the CNN-based single-node VFAA and the convergence simulation experiments of the GVF. The results are shown in Figure 12.

In sub-figure (a) of Figure 12,the initial single node value function of Device 1 is lower than that of Device 2. And it starts to be higher than that of Device 2 at the thousandth update. The single node value function of device 2, finishes converging at the eighth thousandth update. The single node value function of Device 1 stops converging at the ten thousandth update. The highest value of the single node value function of device 1 is 0.0158 and the highest value of the single node value function of device 2 is 0.0015. It can be seen in sub-figure (b) that the GVF starts to converge only at the 500,000th update and then at the two millionth update, and it completes the convergence and the highest value of the GVF of the device is -0.0135. The study on resource scheduling optimization for IOT devices is inadequate as it lacks discussion on the total number of subcarriers involved. It is vital in creating effective scheduling strategies for data transmission, especially in a frequency-division multiplexing system. Their quantity and appropriate allocation are paramount for the seamless and efficient functionality of IOT devices. This can ensure that data transmissions occur without disruptions, interference, or unnecessary congestion. Neglecting the importance of considering the whole number of subcarriers, as the study did, inadvertently compromises the robustness and efficiency of proposed resource scheduling solutions. Finally, the proposed task scheduling optimization algorithm is compared with the current common task scheduling optimization calculation. The results are shown in Table 1.

V. CONCLUSION

The rapid development of the IOT has created a scarcity of computing and storage resources for IOT devices. To reasonably allocate the computing and resources of the devices in the IOT fog computing system, the paper constructs a continuous MDP model. It transforms it into a dynamic optimization solution problem and uses a CNN-based VFAA to solve that. The results indicate that as the number of devices in the IOT fog computing system increases, the algorithm's AD and PC also increase. Among them, the CNN-based VFAA exhibits the lowest rate of increase, with its WS for AD and PC increasing by 0.48, and its increase rate for both algorithms being 0.84. With the increase of the average arrival rate, the AD and PC of the algorithm increase. As the delay weight increases, the AD and PC of the CNN based VFAA gradually decrease, and when the delay weight is 0. With the increase of the delay weight, the AD of CNN-based VFAA gradually decreases. When the delay weight is 0.25, the performance of CNN-based VFAA and multi-user multi-service offloading algorithm is the most reasonable. At this time, the AD of the CNN-based VFAA is about 1S and the PC is about 0.18W. TheAD of the multi-user multi-service offloading algorithm is 1S and the PC is about 0.24W. Therefore, the CNN-based VFAA has the best performance in solving the continuous MDP model.

In the study of resource scheduling optimization for IOT devices, only 12 subcarriers were considered, and the cases of 1, 2, and 6 subcarriers were not considered. If flexible scheduling of subcarriers can be achieved, network coordination can be effectively achieved and the weighted sum of average delay and power consumption can be minimized. Future research may be focused on implementing flexible subcarrier scheduling.

The algorithm proposed in the study can effectively optimize the task scheduling problem in the fog computing system. And enables the continuous Markov decision system to be applied in multi-user systems.

The advantage of the study is that all the conclusions are drawn from objective facts, and all the conclusions of the study are supported by corresponding data. The method proposed by the research solves the limitation of massive data on the development of the IOT, optimizes the utilization efficiency of the IOT devices on computing resources and storage resources. This method is also valuable in guiding task scheduling and decision-making processes within the IOT.

REFERENCES

[1] Q. Huang, Y. Yang, and L. Wang, "Secure data access control with ciphertext update and computation outsourcing in fog computing for Internet of Things," *IEEE Access*, vol. 5, pp. 12941–12950, 2017.
[2] A. Velinov, A. Mileva, S. Wendzel, and W. Mazurczyk, "Covert channels in the MQTT-based Internet of Things," *IEEE Access*, vol. 7, pp. 161899–161915, 2019.
[3] Y. Li, "Construction of U2S communications system based on edge fog computing," *Comput. Commun.*, vol. 153, pp. 569–579, Mar. 2020.
[4] Q. Liu, L. Xiao, J. Yang, and J. C. Chan, "Content-guided convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6124–6137, Sep. 2020.
[5] A. Lakhan, A. A. Abdul Lateef, M. K. Abd Ghani, K. H. Abdulkareem, M. A. Mohammed, J. Nedoma, R. Martinek, and B. Garcia-Zapirain, "Secure-fault-tolerant efficient industrial Internet of Healthcare Things framework based on digital twin federated fog-cloud networks," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 9, Oct. 2023, Art. no. 101747.
[6] M. A. Mohammed, A. Lakhan, K. H. Abdulkareem, and B. Garcia-Zapirain, "Federated auto-encoder and XGBoost schemes for multi-omics cancer detection in distributed fog computing paradigm," *Chemometric Intell. Lab. Syst.*, vol. 241, Oct. 2023, Art. no. 104932.
[7] M. A. Mohammed, A. Lakhan, K. H. Abdulkareem, D. A. Zebari, J. Nedoma, R. Martinek, S. Kadry, and B. Garcia-Zapirain, "Energy-efficient distributed federated learning offloading and scheduling healthcare system in blockchain based networks," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100815.
[8] A. Lakhan, M. A. Mohammed, K. H. Abdulkareem, M. M. Jaber, S. Kadry, J. Nedoma, and R. Martinek, "Fuzzy decision based energy-evolutionary system for sustainable transport in ubiquitous fog network," *Hum.-Centric Comput. Inform. Sci.*, vol. 34, no. 13, pp. 1–15, Jul. 2023.
[9] A. A. Mutlag, M. K. A. Ghani, O. Mohd, K. H. Abdulkareem, M. A. Mohammed, M. Alharbi, and Z. J. Al-Araji, "A new fog computing resource management (FRM) model based on hybrid load balancing and scheduling for critical healthcare applications," *Phys. Commun.*, vol. 59, Aug. 2023, Art. no. 102109.
[10] L. Chen, Z. Zhang, and J. Cao, "A novel method of combining generalized frequency response function and convolutional neural network for complex system fault diagnosis," *PLoS ONE*, vol. 15, no. 2, Feb. 2020, Art. no. e0228324.
[11] G. Pezzano, V. Ribas Ripoll, and P. Radeva, "CoLe-CNN: Context-learning convolutional neural network with adaptive loss function for lung nodule segmentation," *Comput. Methods Programs Biomed.*, vol. 198, Jan. 2021, Art. no. 105792.
[12] Z. Xiong, Q. Guo, M. Liu, and A. Li, "Pan-sharpening based on convolutional neural network by using the loss function with no-reference," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 897–906, Nov. 2021.
[13] P. K. Sharma, N. Gupta, and A. Shrivastava, "Edge enhancement from low-light image by convolutional neural network and sigmoid function," *Smart Moves J. Ijosthe*, vol. 7, no. 1, pp. 8–21, Feb. 2020.
[14] C. Xie and A. Kumar, "Finger vein identification using convolutional neural network and supervised discrete hashing," *Pattern Recognit. Lett.*, vol. 119, pp. 148–156, Mar. 2019.
[15] P. Zhang, Y. Chen, M. Zhou, G. Xu, W. Huang, Y. Al-Turki, and A. Abusorrah, "A fault-tolerant model for performance optimization of a fog computing system," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1725–1736, Feb. 2022.
[16] S. Saraswat, A. Gupta, H. P. Gupta, and T. Dutta, "An incremental learning based gesture recognition system for consumer devices using edge-fog computing," *IEEE Trans. Consum. Electron.*, vol. 66, no. 1, pp. 51–60, Feb. 2020.
[17] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.
[18] I. J. Jacob and P. E. Darney, "Design of deep learning algorithm for IoT application by image based recognition," *J. IoT Social, Mobile, Anal., Cloud*, vol. 3, no. 3, pp. 276–290, Aug. 2021.
[19] J. Huang, S. Wang, G. Zhou, W. Hu, and G. Yu, "Evaluation on the generalization of a learned convolutional neural network for MRI reconstruction," *Magn. Reson. Imag.*, vol. 87, pp. 38–46, Apr. 2022.
[20] T. Li, Y. Xu, J. Luo, J. He, and S. Lin, "A method of amino acid terahertz spectrum recognition based on the convolutional neural network and bidirectional gated recurrent network model," *Sci. Program.*, vol. 2021, pp. 1–7, May 2021.
[21] Z. Gao, H. Zhang, S. Dong, S. Sun, X. Wang, G. Yang, W. Wu, S. Li, and V. H. C. de Albuquerque, "Salient object detection in the distributed cloud-edge intelligent network," *IEEE Netw.*, vol. 34, no. 2, pp. 216–224, Mar. 2020.
[22] F. Li, H. Zhou, Z. Wang, and X. Wu, "ADDCNN: An attention-based deep dilated convolutional neural network for seismic facies analysis with interpretable spatial–spectral maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1733–1744, Feb. 2021.

BING JING was born in December 1979. She received the master's degree major in computer science and technology from Taiyuan Normal University in July. She is currently with the Shaanxi Vocational and Technical College of Finance and Trade. She is also a Lecturer. She mainly engages in the research of computer application technology. So far, she has published five articles and participated in two projects.

HUIMIN XUE was born in June 1980. She received the master's degree major in computer science and technology from Taiyuan Normal University in July. She is currently with the Shaanxi Vocational and Technical College of Finance and Trade. She is also a Lecturer. She mainly engages in the research of computer application technology. So far, she has published six articles and participated in project.