symmetry

Article

Service-Oriented Real-Time Smart Job Shop Symmetric CPS Based on Edge Computing

Chuang Wang [1,2], Yi Lv [2], Qiang Wang [3,*], Dongyu Yang [3] and Guanghui Zhou [4]

[1] Collaborative Innovation Center for Modern Post, Xi'an University of Posts & Telecommunications, Xi'an 710061, China; chuangw@xupt.edu.cn
[2] School of Computer Science & Technology, Xi'an University of Posts & Telecommunications, Xi'an 710061, China; lvyi@stu.xupt.edu.cn
[3] China Electronic Product Reliability and Environmental Testing Research Institute, Guangzhou 510610, China; yangdy@ceprei.com
[4] State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, China; ghzhou@mail.xjtu.edu.cn
* Correspondence: wangqiang@ceprei.com

Abstract: Symmetry is one of the most important notions in the digital twins-driven manufacturing cyber–physical system (CPS). Real-time acquisition of production data and rapid response to changes in the external environment are the keys to ensuring the symmetry of the CPS. In the service-oriented production process, in order to solve the problem of the service response delay of the production nodes in a smart job shop, a CPS based on mobile edge computing (MEC) middleware is proposed. First, the CPS and MEC for a service-oriented production process are analyzed. Secondly, based on MEC middleware, a CPS architecture model of a smart job shop is established. Then, the implementation of MEC middleware and application layer function modules are introduced in detail. By designing an MEC middleware model and embedding function modules such as data cache management, redundant data filtering, and data preprocessing, the ability of data processing is sunk from the data center to the data source. Based on that, the network performances, such as network bandwidth, packet loss rate, and delay, are improved. Finally, an experiment platform of the smart job shop is used to verify different data processing modes by comparing the network performance data such as bandwidth, packet loss rate, and response delay.

Keywords: service-oriented production; cyber–physical systems; mobile edge computing; network performance; real time

check for updates

Citation: Wang, C.; Lv, Y.; Wang, Q.; Yang, D.; Zhou, G. Service-Oriented Real-Time Smart Job Shop Symmetric CPS Based on Edge Computing. *Symmetry* **2021**, *13*, 1839. https://doi.org/10.3390/sym13101839

Academic Editors: Jiewu Leng and Giampaolo Bella

Received: 4 August 2021
Accepted: 10 September 2021
Published: 1 October 2021

1. Introduction

Currently, international competition is becoming increasingly fierce. Improving the intelligence level and rapid response capability of the job shop is a key task to maintain the competitive advantage of manufacturing enterprises. The cyber–physical system (CPS) is an important carrier in the intelligent manufacturing of a smart job shop. It is a multidimensional and complex production system that integrates perception, calculation, and control [1]. It can improve systematic efficiency and the processing of manufacturing resources [2,3]. Fortunately, recent industrial digitalization supports the implementation of the cyber–physical system (CPS).

Symmetry is one of the most important notions in the digital twins-driven manufacturing CPS [4]. Real-time acquisition of production data can not only enrich the cyber space of CPS, but the rapid response of the production system to changes in the external environment can ensure the symmetry of the physical space of CPS [5]. The CPS system obtains the dynamic and static information of the job shop software and hardware resources through the perceptual function of the physical layer. Combined with related application models, the CPS system relies on the powerful computing capabilities of the industrial cloud platform to perform calculation and analysis on a large amount of industrial data. The analysis results are returned to the job shop, and the corresponding action instructions are issued through the control module. However, the CPS system architecture of the job shop based on the cloud platform is facing new requirements. More and more industrial production data need to be sensed, transmitted, and calculated, which creates a time delay for the system control and makes it difficult to control in real time. The intelligent precise control requires rapid calculation and analysis of a large number of data, and the feedback results drive the corresponding operations, which puts forward the requirements for the real-time performance of the production system [6]. In addition, as service-oriented production processes tend to be smaller batches and have shorter cycles, the real-time response to the service requests of each production node greatly affects the whole production cycle and the performance of CPS. Therefore, it is worth studying to improve the performance of CPS by enhancing the real-time response of production nodes.

The CPS facing service-oriented production processes can integrate multilevel distributed manufacturing resources and realize a new manufacturing model for the collaborative production of various manufacturing resources [7,8]. The existing research about the CPS of a service-oriented production process focus on the integration and description of production resources. Most of them improve the performance of the CPS through the combination and scheduling of production service. However, there also are some limitations in current research as follows: (1) poor real-time performance; (2) weak system scalability; (3) lack of security (e.g., data security protection and device access security).

In order to solve these problems, a CPS based on the edge computing is introduced [9–11]. Chen et al. used the edge computing technology to optimize the industrial robot system for thin-film wall welding. They proposed a resource-edge-cloud model that could save 883.38 Kbps bandwidth to meet the demand of industrial products [12]. Zhang et al. applied edge computing to optimize the pumping unit energy-saving control system and improved real-time performance by 80% [13]. Zhang et al. built an abnormal value detection algorithm model based on edge computing, which could effectively detect the data collected by sensors [14]. Sun et al. proposed a video usefulness detection system based on mobile edge computing, which could be used in a smart job shop to judge video content or detect camera faults in real time through edge devices near the video source [15]. Li et al. deployed the computing services of artificial intelligence (e.g., deep learning) to the edge computing devices for the expansion of computing power [16]. Mao et al. verified that edge computing could reduce the response delay on the server side [17]. Edge computing has three advantages in the field of industrial manufacturing: (1) improving the performance of production systems; (2) protecting the data security and privacy; (3) reducing the operation costs and terminal energy consumption. Although edge computing has been widely used in the field of industrial manufacturing, it is still a new computing model that adds computing units to the data source.

The remainder of this article is organized as follows: Section 2 indicates three aspects of the background and motivation of this research. Section 3 describes the CPS architecture for service-oriented production process and edge computing. A CPS model based on mobile edge computing middleware is designed and implemented in Section 4. In Section 5, a smart job shop experiment platform is taken as an example to illustrate the utility of the proposed model. Discussions are presented in Section 6. Section 7 summarizes the principal conclusions of this work and suggests areas of future research.

2. Background and Motivation

2.1. CPS in Smart Job Shop

The cyber–physical system is a core component of the smart job shop [18,19]. It monitors the physical production processes and uses computations and communication deeply embedded in and interacting with physical production processes to add new capabilities to physical systems [3]. Today, the CPS adopted in manufacturing is to increase the production system's openness, autonomy, distributed control, adaptability, and degree of integration. It is able to raise the level of autonomy of production components [20]. CPS is an interconnection of all components (machines and systems) along the value chain, forming a flexible and smart automation system expected to be effective, safe, and efficient for reorganization at run time [21]. The implementation of CPS will lead to significant changes in the working environment, especially in manufacturing and production control systems [22]. It can realize the job shop environment monitoring, production process optimization, product quality inspection, and other aspects of independent decision making and control, and it can also improve production efficiency [23]. Gong et al. designed three types of nonlinear RF chain structures, which reduce the power consumption of massive MIMO systems, and massive multiple-input multiple-output (MIMO) wireless communication technology is an ideal channel to connect the industrial Internet of Things (IIoT) and the CPS [24]. Rathore et al. proposed Deep-Block-IoT Net, and a secure deep learning approach with blockchain for the IoT network is carried out among the edge nodes. The edge layer in a decentralized mode improved the accuracy and reduced the latency of the CPS [25]. Xu et al. studied the description and scheduling of service resources and proposed a hybrid CPS service resource model based on OWL and XML. The task-virtual resource scheduling mechanism solved the problems of low throughput, high drop-out rate, and high delay caused by traditional scheduling strategy [26]. Li et al. built a complex time scheduling algorithm model with uncertain timestamps, which could improve the performance of the production system through the scheduled read–write events in CPS [27].

2.2. Real-Time Information in CPS

As one of the most important characteristics of CPS, real-time performance has also attracted extensive attention from many scholars. Hao et al. proposed an improved deep Q-network (DQN)-based service placement algorithm. The proposed algorithm could achieve an optimal resource allocation by means of convex optimization and reduce the average service response time of CPS by 8–10%. The service placement and workload scheduling decisions were assisted by means of DQN technology [28]. From the perspective of communication technology, Minglei et al. proposed a new communication strategy, which not only improves the real-time performance of the smart grid CPS system but also increases the system throughput [29]. Wang et al. proposed the heterogeneous brain-storming (HBS) method for object recognition tasks in real-world Internet of Things (IoT) scenarios, which enables flexible bidirectional federated learning of heterogeneous models trained on distributed datasets with a new "brain-storming" mechanism and optimizable temperature parameters. It can lower the transmission cost of CPS [30]. In addition, from the perspective of CPS real-time task scheduling, Xu et al. devised a privacy-aware deployment method (PDM) for hosting the machine learning applications in the industrial CPS. The PDM ensured the implementation efficiency of CPS applications and avoided the privacy disclosure of the datasets due to data acquisition by different operators [31]. Zhou et al. proposed a few-shot learning model with Siamese Convolutional Neural Network (FSL-SCNN), which could alleviate the over-fitting issue and enhance the accuracy for intelligent anomaly detection in industrial CPS [32].

2.3. Edge Computing in Manufacturing

At present, it is very urgent in time-sensitive application scenarios to figure out how to avoid massive data flow taking up a large amount of bandwidth resources, relieve the pressure of cloud computing center and link, and improve the real-time performance of the system. Current research studies are focused on edge computing unloading technology and data transmission optimization technology based on edge computing architecture. Tang et al. analyzed the fairness of resource allocation in the cloud environment. They proposed a Yarn-based edge computing resource management platform and a long-term resource fair allocation strategy (LTRF) with an effective data-forwarding vehicle terminal to avoid a large amount of data going to the cloud. However, the data-processing time constraints were not considered in their platform [33]. In the scenario of Internet of

Vehicles based on edge computing, Ning et al. proposed an optimization strategy based on reinforcement learning for a three-layer unloading framework. The framework could satisfy the time-delay constraints of users and minimize power consumption. However, the study only focused on reducing power consumption. It did not take into account the data-intensive application scenarios [34]. Sun et al. gave full consideration to the system energy consumption and delay. They proposed a side-edge calculated unloading cloud architecture model. The optimization of the model was solved by a low-complexity heuristic algorithm. However, the model essentially ignored the cloud computing ability, and it did not distinguish between equipment side, edge, and cloud computing [35]. Li et al. studied a four-layer scheduling model of edge computing. The resource scheduling of the edge layer was mainly realized by a greedy strategy and time-delay constrained threshold strategy. The artificial intelligence task operation was realized from the perspective of the network. The real-time performance of the system can be effectively ensured [36]. Combining blockchain technology and edge computing, Xu et al. proposed a new unloading method. The integrity of task data transmission in the process of task unloading was solved by a non-dominated sorting genetic algorithm [37]. However, its solution procedure is relatively not efficient. It is not suitable for computation-intensive scenarios.

3. The Mobile Edge Computing Architecture in Job Shop

3.1. CPS for Service-Oriented Production Process

The CPS for the service-oriented production process is based on the Internet environment, which encapsulates all kinds of manufacturing resources (e.g., hardware and software resources) in the job shop into a series of services. The unified CPS service management platform can realize reusability and interoperability among heterogeneous manufacturing nodes, systems, and development platforms. Compared with the existing embedded real-time system and network control system, the CPS pays more attention to the real-time perception and dynamic supervision of information resources and physical resources in the production process. It also provides more flexible, real-time, and efficient services and management for each production node. The specific production process can be abstractly described, as Figure 1 shows.

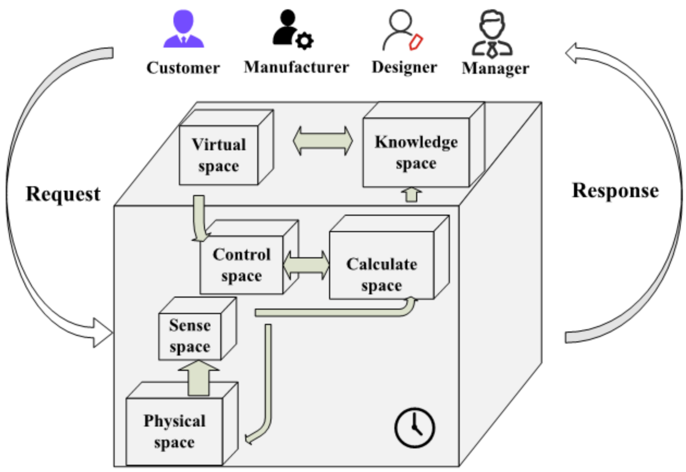

Figure 1. The black box model for service-oriented production process.

As shown in Figure 1, CPS is abstracted into a black box model, in which the heterogeneity of production nodes, systems, and development platforms is shielded by some new features (e.g., reusability, loosely couple, abstraction, transparency). When each production node sends a resource service request to the CPS data center, every intermediate link is hidden, and it only needs to care about whether it responds in a timely manner.

3.2. The Mobile Edge Computing

The mobile edge computing (MEC) is a new computing architecture model, unlike the cloud computing architecture. As shown in Figure 2, the MEC middleware is added between the cloud computing center and the mobile terminal devices to provide the end users with resource services (e.g., converged computing, storage, and network). It is equivalent to moving some functions of the cloud computing center to the edge. First of all, the user equipment (UE) submits resource service requests to the nearest MEC middleware through the evolved Node-B (eNB). Then, the MEC middleware replaces the data center to process resource service requests. Meanwhile, edge computing is also an enabling technology. It can provide resources on the edge of the network to meet some key requirements such as agile connections, real-time services, data optimization, application intelligence, security, and privacy protection.

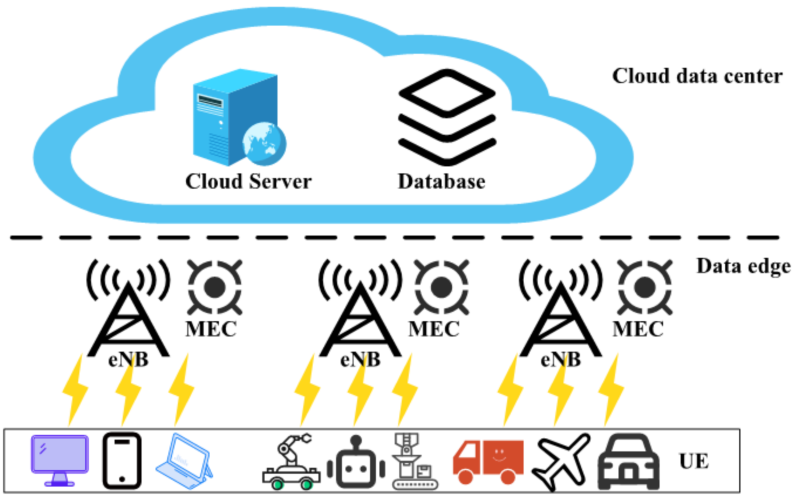

Figure 2. MEC architecture model.

For the service-oriented production process, the MEC architecture has at least the following three outstanding advantages.

(1) Real time. The computing tasks in the cloud computing center are partially or totally unloaded to the network edge (such as nearby terminal device and data source) without transmitting a large number of unprocessed data. It can effectively reduce the computing load of the cloud computing center and optimize the CPS network performance to ensure the real-time processing of data.

(2) Security. Under the MEC architecture, a large number of data are not only processed at the terminal device or data source but also backed up by multiple nodes. It avoids the data loss caused by a large amount of data transmission or cloud computing center fault.

(3) Scalability. The MEC architecture can provide a scalable and cheaper method, which allows functional expansion through embedded technology. It expands its capabilities of data computing and processing through a combination of edge devices.

4. Smart Job Shop CPS Based on MEC Middleware

In this article, the smart job shop CPS is defined as a multiple job shops production system. Each job shop includes different types of production nodes (such as workpieces, AGV, industrial robot, fixtures tools, cutting tools, measuring tools, sensors, operator, etc.). All data transmission, for example production monitoring data, resources request data, etc., is also defined as a service in the production process.

4.1. Smart Job Shop CPS Architecture Model

According to the characteristics of the MEC architecture model, MEC middleware is added to the traditional CPS model, and the performance of the CPS network is optimized to ensure the real-time response to the service requests of each production node. The CPS architecture model based on MEC middleware is described in Figure 3.

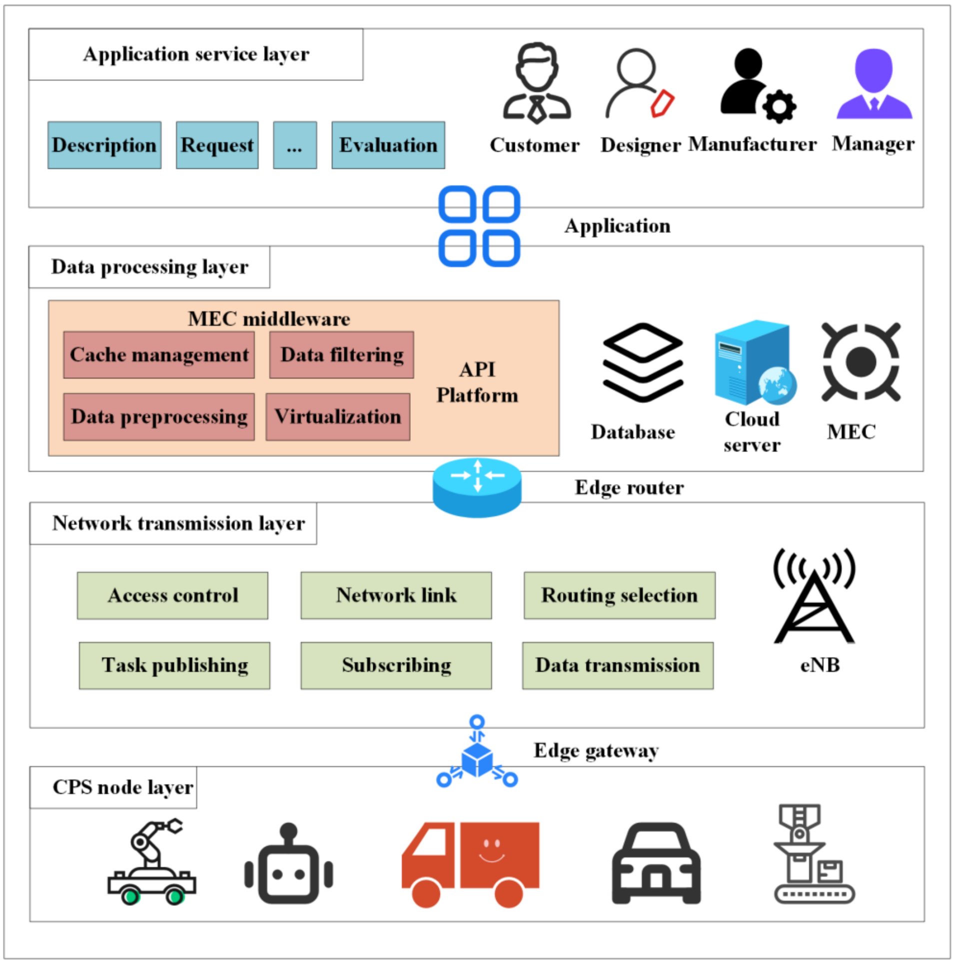

Figure 3. CPS architecture model based on MEC middleware.

In the service-oriented production process, CPS provides various functions to end users (i.e., production nodes) in the form of services. Each production node is both a service requester and a service receiver. As shown in Figure 3, the CPS architecture of the service-oriented production process can be divided into four layers from bottom to top, namely node layer, network transmission layer, data processing layer, and application service layer.

(1) CPS node layer. The CPS node layer is a real physical production entity in the physical job shop; it includes all manufacturing resources such as the CNC lathe, industrial robot, AGV, cutting tool, fixture tool, measuring tool, sensors, PLC, operator, etc. This layer is the most important layer in the CPS architecture because it realizes the interaction between the production system and the physical world, and it reflects the collaboration of the informational process and physical process.

(2) Network transmission layer. This layer connects various remote resources through Ethernet access and provides resource services for the entire production system. Therefore, it is the basis of resource sharing in the whole production system of the smart job shop. It can provide some basic functions such as data access control, network link, routing selection, task publishing and subscribing, data transmission, etc.

(3) Data processing layer. This layer is the core part of the CPS model, which realizes the transfer of data processing power through MEC middleware. The MEC middleware can expand its functions through an application programming interface (API) to complete data

cache management, redundant data filtering, data preprocessing, etc. The implementation of MEC middleware is detailed as described in the following sections.

(4) Application service layer. The resource capabilities of CPS, such as perception, execution, calculation, communication etc., are abstracted into a series of services at this layer. The managers of the smart job shop can monitor the interaction between production nodes and CPS through the interactive platform, including interactive content such as service description, service request, service query, service response, service evaluation, etc.

4.2. The Implementation of MEC Middleware

The MEC middleware includes three logical entities: the MEC infrastructure layer, MEC platform layer, and MEC application layer, as Figure 4 shows. Furthermore, it also contains an API that can extend other functions.

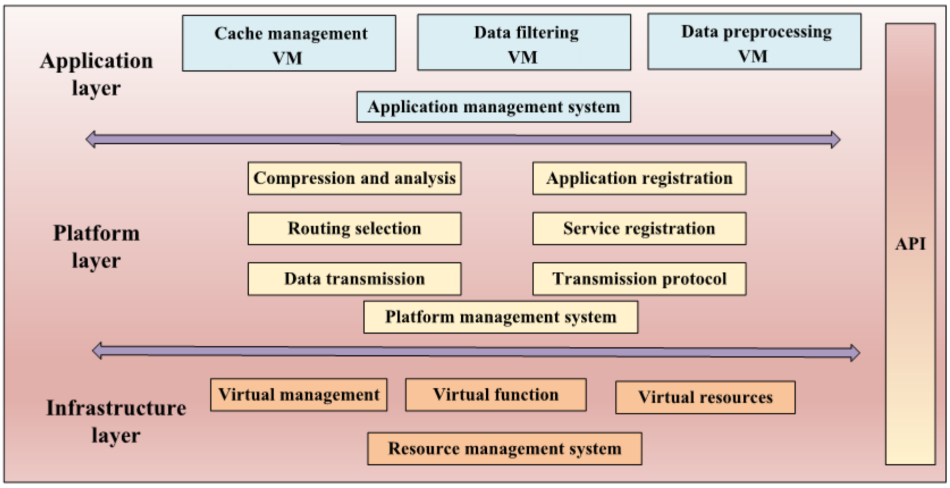

Figure 4. The functions of MEC middleware model.

(1) Infrastructure layer. The MEC infrastructure layer is based on hardware resources such as the server, microprocessor, software-defined network (SDN) controller, etc. At the same time, these are also carriers of middleware implementation. The layer uses virtual technology to provide a virtualized data computing function, caching, virtual exchange function, and other virtual functions under the resource management system. It is an important support to realize the sinking of data computing and processing functions from the cloud data center to the data source.

(2) Platform layer. The MEC platform layer regards the functions provided by the infrastructure layer as a series of services under the platform management system. In addition, this layer also provides the upper system with a flexible and efficient platform environment that includes some basic function modules such as data packet compression and analysis, content routing, wireless data interaction, application registration management, service registration management, transmission protocol optimization, etc.

(3) Application layer. Based on the application management system, the basic functions of the MEC application layer provided by the MEC platform are further encapsulated as some virtual applications. More application modules that depend on different MEC platforms (e.g., data cache management, redundant data filtering, data preprocessing, etc.) can be embedded through standard API. This can ensure that the middleware has very convenient scalability.

The scenario configuration of MEC middleware in a smart job shop CPS is shown in Figure 5. Each job shop is equipped with an MEC server to form a two-level interactive architecture of an enterprise data center and job shop MEC. Through embedded development technology, the algorithm modules such as the data cache management module (e.g., Least Frequently Used algorithm), redundant data-filtering module (e.g., Bloom Filer algorithm), and data preprocessing module (e.g., machine learning algorithms) are embedded to realize the data preprocessing function. Furthermore, resource sharing can be implemented in different job shops through the job shop MEC, and production nodes in different job shops are interconnected.

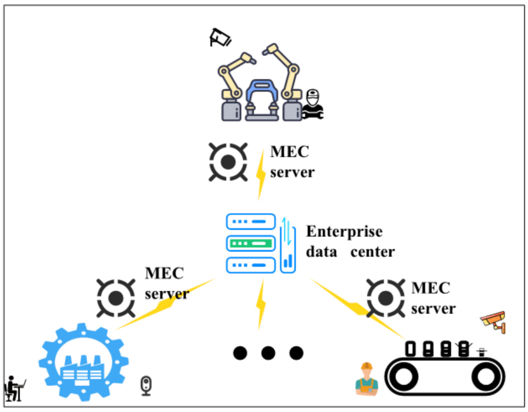

Figure 5. The Scenario configuration of MEC middleware.

4.3. The Realization of the Application Layer Function Module

(1) Data cache management

The function module of data cache management is embedded in the job shop MEC server. The Least Frequently Used (LFU) algorithm is used to evaluate the value of the data to realize the cache management. Assumption: the size of the cache is defined as N, and the cached data are $d_1, d_2, \ldots \ldots, d_n$. The worth of the data can be calculated as follows.

$$R(d_n) = L + f(d_n) \times c \ (0 < c < 1) \tag{1}$$

where $R(d_n)$ is used to evaluate the worth of the data, $f(d_n)$ represents the data access frequency, L is an expansion factor, and $f(d_n) \times c$ makes the access data at different times with different weights.

The MEC server of the job shop calculates the worth value (also called as the hot degree) of the job shop production data captured by sensors including the layout information of job shop machine tools, product quality information, manufacturing resource allocation information, etc., as Figure 6 shows. Then, the data with larger hot degree values are cached into the MEC server in advance to replace the data with the lower hot degree value in storage. When each production node sends a production resource service request, the data cache management prioritizes data association and matching from the cache of the MEC server. This function module achieves the timely response of resource service requests at the local job shop.

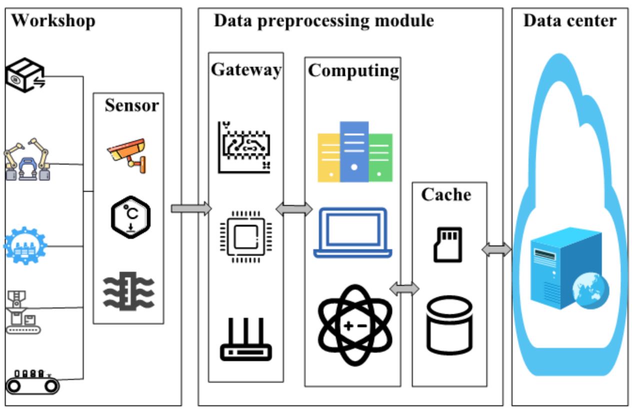

Figure 6. The implementation of data cache management.

(2) Redundant data filtering

When data association and matching fail from the cache list of the MEC server, a large amount of redundant service request data would be sent to the enterprise data center. This would occupy a large amount of bandwidth resources and easily cause greater access pressure to the data center. As a result, the service request response is not timely. Therefore, the redundant data filtering of the function module is very important for the MEC server of the job shop.

The Bloom filter is adopted into the implementation of the redundant data filtering. Assumption: the misjudgment rate is p. It consists of an m-bit array initialized to 0 and k independent hash functions, $h_1, h_2, h_3 \ldots\ldots h_k$. A data flow set, $S = d_1, d_2, d_3 \ldots\ldots d_n$, is firstly set. All new data elements of the data set are mapped to different positions in the array through each hash function. If the corresponding position is 0, it is set to 1. Finally, if all positions of the array are 1, it means that this data element is already in the data set S and the data are redundant. The Bloom filter can be described as following:

$$\begin{cases} p = \left(1 - e^{-\frac{nk}{m}}\right)^k \\ k = \ln 2 \times \frac{m}{n} \\ m = -\frac{n \times \ln p}{(\ln 2)^2} \end{cases} . \quad (2)$$

In a Bloom filter with a misjudgment rate p, the position of all hash values of data elements in the array is 0 or 1, which determines whether they are redundant data. Therefore, according to the position, the redundant data can be filtered and processed.

(3) Data preprocessing

On the data source side, the function module of data preprocessing is embedded in the MEC server to process a large amount of production data at the local job shop. In the data preprocessing module, some machine learning algorithms embedded in the MEC server can realize data intelligent processing so that a large amount of unprocessed production data can be processed at the data source without occupying a large amount of bandwidth resources for unnecessary data transmission.

5. Case Study

In order to verify the proposed CPS model, a smart manufacturing platform for a job shop for a turbine blade is taken as an example. The network performance of the job shop (such as network bandwidth, packet loss rate, and server access delay) is tested under two different data processing modes, namely data center processing and MEC processing.

5.1. Experimental Environment

The layout of the smart manufacturing platform of job shop is depicted in Figure 7, which mainly consists of the following hardware and software: a GL8-V five-axis machining center (VMC850 CNC), machining center (FT-20 CNC lathe), industrial robot, AGV, multi-station silo, RFID devices, MES, materiel storage management system, etc. A GL8-V five-axis machining center is a machining center for complex surfaces, which is equipped with a wheel-disc automatic and rotatable tool library. A VMC850 CNC machining center is a CNC milling machine equipped with a wheel-disc automatic tool library, which can finish milling and drilling processes. An FT-20 CNC lathe is an automatic milling equipment with a cutter head based on a pre-programmed program. All production equipment is equipped with an RFID reader near its tool library, which is used to output the tool information of different tool locations.

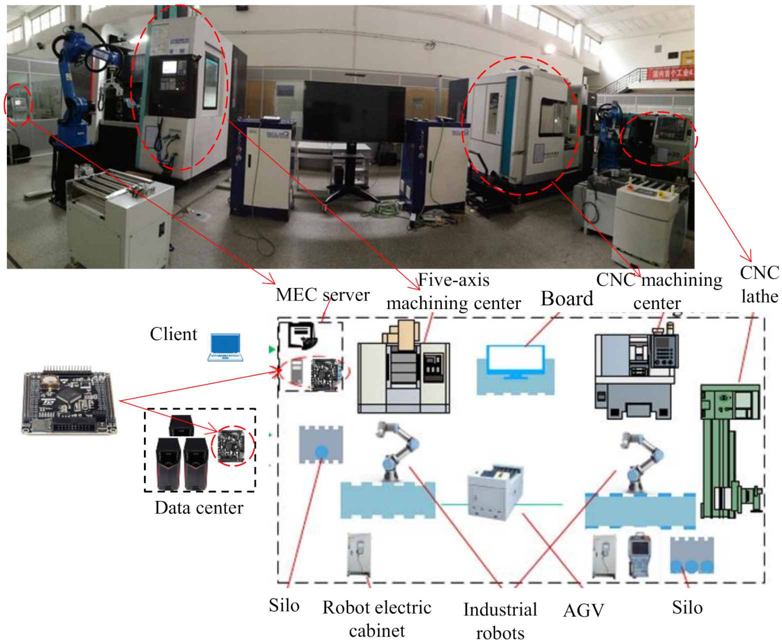

Figure 7. The layout of the experimental environment.

Figure 7 illustrates the prerequisites of the testbed, which involves a data center, MEC server deployed at the edge, manufacturing cell, and client. As for the building blocks of the manufacturing cell, it consists of two industrial robots, a robot electric cabinet, two silos, a five-axis machining center, a CNC machining center, a CNC lathe, an AGV, and a Kanban system. In addition, IoT devices are used to perceive the real-time status of the manufacturing cell. For example, sensors deployed on the industrial robots and machine tools such as temperature sensors, speed sensors, and liquid level sensors perceive the linear displacement, angular displacement, temperature, pressure, and velocity of these devices.

In addition, the server with the STM32F407 development board is configured as the job shop MEC server. The LFU algorithm, Bloom filter algorithm, and machine learning algorithm are used as the data cache management module, redundant data filtering module, and data preprocessing module, respectively. The network server of the job shop is regarded as the enterprise data center. A notebook computer is configured with a 2.4 GHz Intel Core i5, memory 16 GB 2133 MHz LPDDR3, and MACOS10.14.6 as the agent client of the job shop, which is used to connect the MEC server and data center.

5.2. Experimental Verification and Results

In the process of turbine blade machining, tool monitoring data are output as the test data by RFID embedded in a five-axis machining center.

The experiment is divided into several steps. Firstly, the test data are encapsulated into a large number of UDP data packets, and then, they are transmitted to simulate the service requests of different orders of magnitude by setting the parameters of Iperf software. Then, the Wireshark package capture software is used to monitor and capture UDP data packets and analyze the three key network performances such as bandwidth, packet loss rate, and delay.

Different fixed-size UDP data packets, such as 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, and 100, are sent to the MEC server and data center by setting the parameters of Iperf in agent client, which simulate the data transmission of different orders of magnitude service requests in turn. The results are shown in Figure 8.

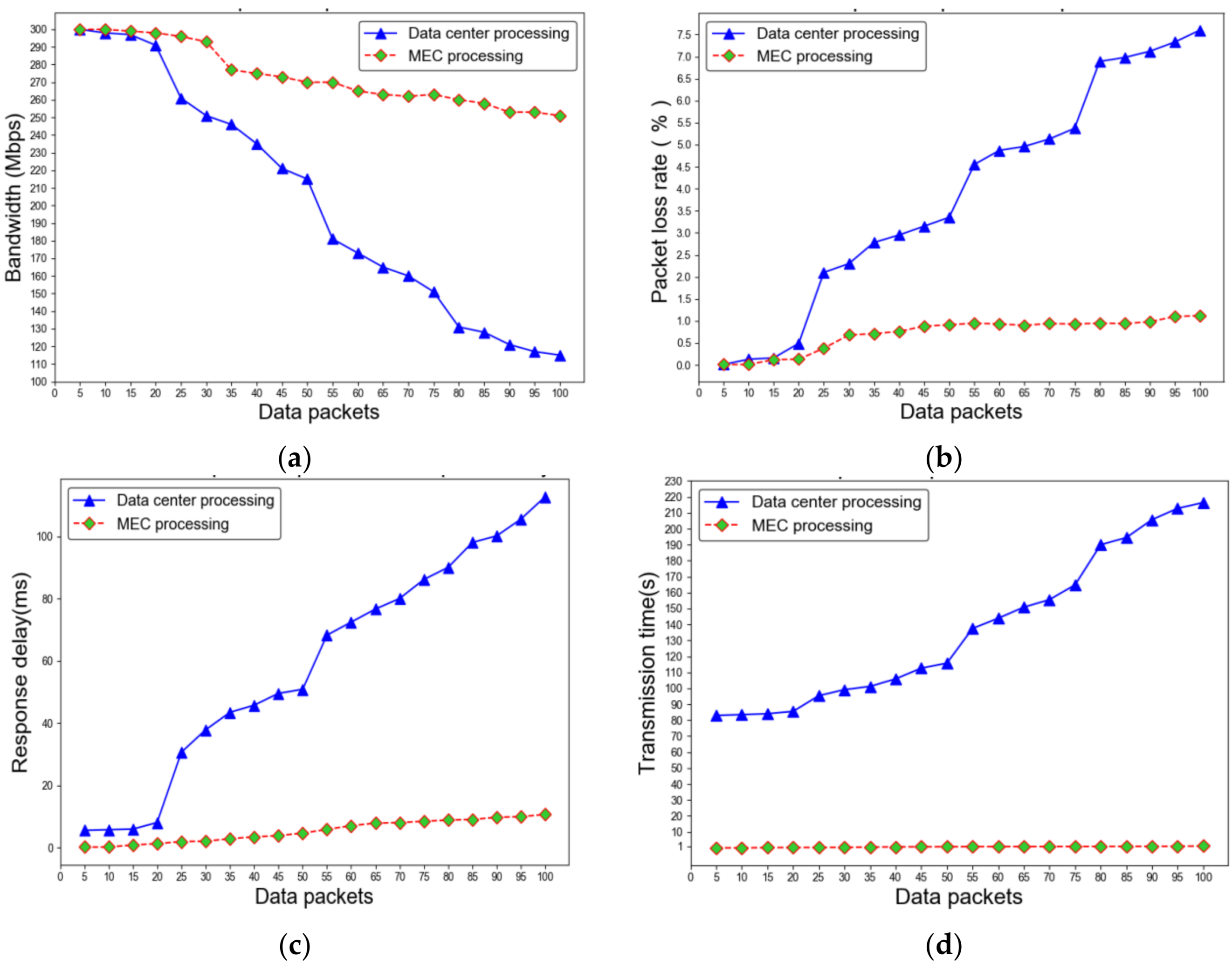

Figure 8. The experiment results of different fixed size UDP data packets. (**a**) the relationship of data packets and bandwidth; (**b**) the relationship of data packets and packet loss rate; (**c**) the relationship of data packets and response delay; (**d**) the relationship of data packets and transmission time.

6. Discussion

6.1. Result Analysis

(1) The impact of data packet transmission on network bandwidth

As Figure 8a shows, the bandwidth is reduced to 63.3% with the increase in the number of data packets transmission under the traditional data center processing mode. However, under the MEC processing mode, the bandwidth is only reduced by 10%. The reduction of bandwidth would directly affect the network performance of the job shop and reduce the stability and speed of data transmission in CPS.

(2) The impact of data packet transmission on packet loss rate

With the increase in the number of data packets transmission, the packet loss rate rises in a fold line under the traditional data center processing mode. As Figure 8b shows, in the three stages (20–25, 50–55, 75–80), the packet loss rate rises sharply due to the significant decrease in bandwidth, and it finally reaches 7.56%. Under the MEC processing mode, the packet loss rate is always within 1%, showing good network stability, which can guarantee the stability of data transmission in the CPS.

(3) The impact of data packets transmission on delay

Figure 8c shows that when the MEC server performs data preprocessing, the response delay of the data center server is as high as 11 milliseconds, and the response speed is very fast. However, the traditional CPS does not have the data preprocessing capability, and the response delay reaches 168 milliseconds. The server response delay directly determines whether CPS can respond to the service request of each production node in real time.

(4) The impact of network environment on data transmission time

In different network environments, the MEC server can preprocess 3 GB-sized data and transmit the resulting data to the data center within 1 s. Compared with the traditional data transmission method, the time is greatly reduced. Therefore, CPS based on the MEC server can not only avoid bandwidth resources occupied by a large amount of data transmission but also improve the efficiency of data processing.

6.2. Challenges

Although the model guarantees the real-time response of each production node by optimizing the CPS network performance, there are still some problems as follows: (1) The data transmission distance and data processing time of the MEC server are ignored during the experiment; (2) During the experiment, the service requests of each production node are simulated through fixed-size UDP data packets. While in the actual production process, the size of service request data packets is dynamic and unstable; (3) Compared with the real production job shop, the experimental environment is more stable and simpler. Therefore, verifying the universality of the CPS model in the complex production job shop environment is the next research direction.

7. Conclusions

In order to solve the response lag problem of a service-oriented production process, this article proposes a job shop CPS model based on MEC middleware. After the in-depth analysis of the current job shop CPS system, the edge cloud collaborative computing is innovatively introduced. The job shop CPS system architecture includes a CPS node layer, network transmission layer, data processing layer, and application service layer. The data processing layer and application service layer reshape the theory of the job shop CPS system through the combination of edge computing technology and cloud computing technology. Furthermore, the functions of the MEC middleware model are analyzed from the infrastructure layer, platform layer, and application layer. Then, the realization of the application layer function module is divided into data cache management, redundant data filtering, and data preprocessing.

By comparing the data processing mode of the new CPS model with the traditional CPS, the results show that the bandwidth decreases by 63.3%, the packet loss rate keeps within 1%, and the response delay of the data center lasts within 11 milliseconds. It can be seen that it is feasible to optimize the network performance of CPS by data preprocessing from the data source, so as to improve the real-time response of CPS.

In future work, how to implement edge decision making based on the CPS model and how to improve the real-time response and performance of CPS through edge intelligent decision making need to be deeply studied.

Author Contributions: Conceptualization, C.W.; funding acquisition, Q.W.; investigation, D.Y.; writing—original draft preparation, Y.L.; writing—review and editing, G.Z.; validation, Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key R&D Program of China under Grant 2018 YFB1802403, China Postdoctoral Science Foundation under Grant 2018M643727, Soft Science Research Project of Shanxi Province under Grant 2019KRM122, Special scientific research plan of Education Department of Shaanxi Provincial Government under Grant 20JK0922, National Natural Science Foundation of China under Grant 51975463, the APC was funded by 2018YFB1802403.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Y.; Peng, Y.; Wang, B.; Yao, S.; Liu, Z. Review on cyber-physical systems. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 27–40. [CrossRef]
2. Panetto, H.; Iung, B.; Ivanov, D.; Weichhart, G.; Wang, X. Challenges for the cyber-physical manufacturing enterprises of the future. *Annu. Rev. Control* **2019**, *47*, 200–213. [CrossRef]
3. Wang, L.; Törngren, M.; Onori, M. Current status and advancement of cyber-physical systems in manufacturing. *J. Manuf. Syst.* **2015**, *37*, 517–527. [CrossRef]
4. Leng, J.; Zhang, H.; Yan, D.; Liu, Q.; Chen, X.; Zhang, D. Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart job shop. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1155–1166. [CrossRef]
5. Leng, J.; Jiang, P.; Liu, C.; Wang, C. Contextual self-organizing of manufacturing process for mass individualization: A cyber-physical-social system approach. *Enterp. Inf. Syst.* **2020**, *14*, 1124–1149. [CrossRef]
6. Guo, J.; Yang, Z.; Chen, C.; Luo, W.; Hu, W. Real-Time Prediction of Remaining Useful Life and Preventive Maintenance Strategy Based on Digital Twin. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 1–48. [CrossRef]
7. Esp-Beltrn, J.V.; Gilart-Iglesias, V.; Ruiz-Fernndez, D. Enabling distributed manufacturing resources through SOA. *Robot. Comput.-Integr. Manuf.* **2017**, *46*, 156–165. [CrossRef]
8. Herwan, J.; Kano, S.; Oleg, R.; Sawada, H.; Kasashima, N. Cyber-physical system architecture for machining production line. In Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, Russia, 15–18 May 2018; pp. 387–391.
9. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
10. Weisong, S.; Hui, S.; Jie, C.; Quan, Z.; Wei, L. Edge computing-An emerging computing model for the Internet of everything era. *J. Comput. Res. Dev.* **2017**, *54*, 907–924. (In Chinese)
11. Mach, P.; Becvar, Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1628–1656. [CrossRef]
12. Chen, Y.; Hu, J. Industrial application based on edge computing: AGV control system. *Comput. Integr. Manuf. Syst.* **2019**, *25*, 3191–3198. [CrossRef]
13. Zhang, S. Application of edge calculation in energy-saving control of beam pumping unit. *Autom. Expo.* **2019**, *36*, 54–57.
14. Zhang, Q.; Hu, Y.; Ji, C.; Zhan, P.; Li, X. Edge Computing Applications: Real-time Sensor Data Anomaly Detection Algorithms. *Comput. Res. Dev.* **2018**, *55*, 524–536.
15. Sun, H.; Liang, X.; Shi, W. VU: Video usefulness and its application in large-scale video surveillance systems: An early experience. In Proceedings of the Workshop on Smart Internet of Things, New York, NY, USA, 14 October 2017; pp. 1–6.
16. Li, H.; Ota, K.; Dong, M. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Netw.* **2018**, *32*, 96–101. [CrossRef]
17. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
18. Wang, C.; Zhou, G.; Zhu, Z. Service perspective based production control system for smart job shop under industry 4.0. *Robot. Comput.-Integr. Manuf.* **2020**, *65*, 101954. [CrossRef]
19. Uhlemann, H.J.; Lehmann, C.; Steinhilper, R. The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0. *Procedia CIRP* **2017**, *61*, 335–340. [CrossRef]
20. Gronau, N.; Grum, M.; Bender, B. Determining the optimal level of autonomy in cyber-physical production systems. In Proceedings of the 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, France, 19–21 July 2016; pp. 1293–1299.
21. Mordinyi, R.; Biffl, S. Versioning in Cyber-physical Production System Engineering—Best-Practice and Research Agenda. In Proceedings of the 2015 IEEE/ACM 1st International Workshop on Software Engineering for Smart Cyber-Physical Systems, Florence, Italy, 17 May 2015; pp. 44–47.
22. Graessler, I.; Poehler, A. Integration of a digital twin as human representation in a scheduling procedure of a cyber-physical production system. In Proceedings of the 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 10–13 December 2017; pp. 289–293.
23. Pooranian, Z.; Shojafar, M.; Garg, S.; Taheri, R.; Tafazolli, R. LEVER: Secure Deduplicated Cloud Storage With Encrypted Two-Party Interactions in Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5759–5768. [CrossRef]
24. Gong, Y.; Zhang, L.; Liu, R.P.; Yu, K.; Srivastava, G. Nonlinear MIMO for Industrial Internet of Things in Cyber–Physical Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5533–5541. [CrossRef]
25. Rathore, S.; Park, J.H. A Blockchain-Based Deep Learning Approach for Cyber Security in Next Generation Industrial Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5522–5532. [CrossRef]
26. Xu, J.Q.; Guo, X.J.; Wang, J.; Li, H.; Zhao, H. Research on CPS Resource Service Model and Resource Scheduling. *J. Comput. Sci.* **2018**, *41*, 2330–2343. [CrossRef]

27. Fangfang, L.; Chong, L.; Ge, Y. Scheduling algorithm of events with imprecise timestamps for CPS. *J. Front. Comput. Sci. Technol.* **2017**, *11*, 887896. (In Chinese)
28. Hao, Y.; Chen, M.; Gharavi, H.; Zhang, Y.; Hwang, K. Deep Reinforcement Learning for Edge Service Placement in Softwarized Industrial Cyber-Physical System. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5552–5561. [CrossRef]
29. You, M.; Liu, Q.; Sun, H. New communication strategy for spectrum sharing enabled smart grid cyber-physical system. *IET Cyber-Phys. Syst. Theory Appl.* **2017**, *2*, 136–142. [CrossRef]
30. Wang, C.; Yang, G.; Papanastasiou, G.; Zhang, H.; Rodrigues, J.J.P.C.; de Albuquerque, V.H.C. Industrial Cyber-Physical Systems-Based Cloud IoT Edge for Federated Heterogeneous Distillation. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5511–5521. [CrossRef]
31. Xu, X.; Mo, R.; Yin, X.; Khosravi, M.R.; Aghaei, F.; Chang, V.; Li, G. PDM: Privacy-Aware Deployment of Machine-Learning Applications for Industrial Cyber–Physical Cloud Systems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5819–5828. [CrossRef]
32. Zhou, X.; Liang, W.; Shimizu, S.; Ma, J.; Jin, Q. Siamese Neural Network Based Few-Shot Learning for Anomaly Detection in Industrial Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5790–5798. [CrossRef]
33. Tang, S.; Lee, B.S.; He, B. Fair resource allocation for data-intensive computing in the cloud. *IEEE Trans. Serv. Comput.* **2016**, *11*, 20–33. [CrossRef]
34. Ning, Z.; Dong, P.; Wang, X.; Guo, L.; Rodrigues, J.J.; Kong, X.; Huang, J.; Kwok, R.Y.K. Deep reinforcement learning for intelligent Internet of Vehicles: An energy-efficient computa-tional offloading scheme. *IEEE Trans. Cogn. Commun. Netw.* **2019**, *5*, 1060–1072. [CrossRef]
35. Sun, C.; Hui, L.; Li, X.; We, J.; Xiongl, Q.; Wang, X.; Leun, V.C. Task offloading for end-edge-cloud orchestrated computing in mobile networks. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea, 25–28 May 2020; pp. 1–6.
36. Li, X.; Wan, J.; Dai, H.N.; Imran, M.; Xia, M.; Celesti, A. A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4225–4234. [CrossRef]
37. Xu, X.; Zhang, X.; Gao, H.; Xue, Y.; Qi, L.; Dou, W. BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4187–4195. [CrossRef]