Engineering 5 (2019) 662–670

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Intelligent Manufacturing—Article

Fog-IBDIS: Industrial Big Data Integration and Sharing with Fog Computing for Manufacturing Systems

Junliang Wang [a], Peng Zheng [b], Youlong Lv [a], Jingsong Bao [a], Jie Zhang [a,*]

[a] College of Mechanical Engineering, Donghua University, Shanghai 201620, China
[b] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 22 May 2018
Revised 31 August 2018
Accepted 5 December 2018
Available online 5 July 2019

Keywords:
Fog computing
Industrial big data
Integration
Manufacturing system

ABSTRACT

Industrial big data integration and sharing (IBDIS) is of great significance in managing and providing data for big data analysis in manufacturing systems. A novel fog-computing-based IBDIS approach called Fog-IBDIS is proposed in order to integrate and share industrial big data with high raw data security and low network traffic loads by moving the integration task from the cloud to the edge of networks. First, a task flow graph (TFG) is designed to model the data analysis process. The TFG is composed of several tasks, which are executed by the data owners through the Fog-IBDIS platform in order to protect raw data privacy. Second, the function of Fog-IBDIS to enable data integration and sharing is presented in five modules: TFG management, compilation and running control, the data integration model, the basic algorithm library, and the management component. Finally, a case study is presented to illustrate the implementation of Fog-IBDIS, which ensures raw data security by deploying the analysis tasks executed by the data generators, and eases the network traffic load by greatly reducing the volume of transmitted data.

1. Introduction

With the development of the Internet of Things (IoT), multiple sensors, and other data-sensation technology, an exponentially growing volume of data is being captured in industry practices [1,2]. This captured big data can help manufacturers to improve their production efficiency [3,4], system resilience and sustainability [5,6], and product quality [7], and to achieve better customer experience through precision marketing and design. Moreover, the models [8], approaches [9,10], and platforms [11] that have emerged enable big data to be used to achieve dramatic improvements in the operation of complex manufacturing systems [12,13]. Hence, big data analytics is now becoming prevalent and is expected to become a widely used method in the operation of complex manufacturing systems [14]. Communicating, aggregating, storing, analyzing, and visualizing collected data are now all part of the operation of an advanced manufacturing system [15], as realized through the cyber–physical system (CPS). Within a CPS, data, information, and knowledge are combined to form an integrated environment for system optimization. Data integration is the basis for system integration, as data serves as the information and knowledge of a CPS [16]. Thus, industrial big data integration and sharing (IBDIS) determines the efficiency of big data analysis and plays a key role in the operation of manufacturing systems. IBDIS provides massive industrial data that can be analyzed by defining, extracting, transforming, and loading [17]. However, big data integration in manufacturing systems remains a difficult task due to two challenges: heavy network traffic load and privacy concerns regarding original industrial big data.

1.1. Heavy network traffic load

The first challenge is the heavy network traffic load that results from the long-distance transmission of massive raw data for big data analysis. In modern manufacturing systems, the number of intelligent machines and IoT devices generating massive data has been growing, so as to empower Industry 4.0 [18]. The volume of data generated by sensors embedded in machine tools, cloud-based solutions, and business management has already reached more than 1000 EB annually, and is expected to increase in the years to come [19]. As a result, expansion of an industrial cloud-based data center is required in order to support data integration and storage for the operation of manufacturing systems [17,20].

* Corresponding author.
E-mail address: mezhangjie@dhu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.eng.2018.12.013

However, the cloud-computing framework transmits massive raw data into a remote cloud-based data center for further analysis [21,22], which results in a heavy network traffic load. Industrial big data has the characteristic of being multi-sourced, which means that the collected data (containing order requirements, product process routes, machine statuses, and plan solutions) comes from product data management (PDM) systems, manufacturing execution systems (MES), supervisory control and data acquisition (SCADA) systems, and so forth [23]. Hence, multiple data exchanges occur frequently during IBDIS, increasing the network traffic load. At present, existing commercial cloud-computing models upload original industrial big data using a batch-processing mode (once/twice per day at midnight) in order to ease the network load; however, this cannot satisfy the requirement of real-time optimization in the industrial field.

1.2. Privacy concerns regarding original industrial big data

The second challenge involves the privacy of original industrial big data, as shared data can be copied and secondary traded by the customer during data exchange. In industry, big data is usually classified and is only accessible within the local company network [24]. In the majority of manufacturing organizations, raw data is not allowed to be transmitted to a remote commercial cloud center directly, since private raw data can easily be copied during storage and transmission. In some high-technology enterprises, data is the core of the company's assets, and data leakage is likely to bring incalculable losses to the enterprise. For example, the recipes, customer data, and process parameters are closely guarded in wafer manufacturing, as they are vital to a semiconductor foundry. Therefore, the privacy of original industrial big data should be carefully considered in IBDIS.

To address these two issues, this paper puts forward a fog-computing architecture named Fog-IBDIS, which takes full advantage of the computing power of edge devices in the network to preprocess the original industrial big data. In Fog-IBDIS, the data integration and sharing tasks are moved to the edge devices, such as industrial personal computers and application servers. With Fog-IBDIS, the volume of data is reduced during preprocessing in order to improve the time latency and network traffic load in the integration of big data. In addition, data that has been processed through standardization or other data-transformation technologies is far less sensitive for enterprises than raw data. With the proposed Fog-IBDIS, the intermediate result is transmitted and uploaded in the big data analytical task in order to protect the privacy of the original data.

The rest of this article is structured as follows. First, related studies about big data integration and fog computing are reviewed. Next, Fog-IBDIS is proposed, with a task flow graph (TFG) and a schematic diagram, as an approach to model and manage the integration process of industrial big data. The function of Fog-IBDIS is then organized into five modules to realize the integration and sharing of industrial big data. Subsequently, a case study is performed to illustrate the implementation and performance of Fog-IBDIS, and the differences between IBDIS with cloud computing and Fog-IBDIS are discussed. Finally, the conclusion and future directions are detailed.

2. Related works

Since industrial big data is characterized by the "three Vs" of volume, variety, and velocity, and by the "three Ms" of multi-source, multi-dimension, and multi-noise, big data integration must extract and transform raw data with complex schemas in order to manage and organize the massive data. This process is of great importance, and has received a significant amount of consideration in past decades. Xiang et al. [25] proposed a hybrid manufacturing cloud architecture to centralize and share manufacturing data in the product life-cycle. This designed big data integration approach is supported and promoted by a flexible private cloud-computing platform. Ma'ayan et al. [26] developed a centralized data management method to integrate the data from large-scale projects in systems biology and systems pharmacology into a single unified data pool. Mezghani et al. [27] designed a generic semantic big data architecture to manage the diversity and variety of wearable data related to healthcare. They also implemented and evaluated a wearable Kaas platform to smartly manage and centralize heterogeneous data coming from wearable devices in order to assist physicians in supervising patient health evolution and to keep the patient up to date about his or her status. In the data management of IoT data, Jiang et al. [28] proposed a data storage framework that involved combining multiple databases with a Hadoop platform in order to store and manage diverse types of data collected by massive IoT devices. With the cloud-computing platform, rapidly generated IoT data could be stored and processed effectively. Chang et al. [29] integrated several kinds of big data warehouse platforms, and selected the best one to manage big datasets. With this optimized big data platform, the data could be centralized and processed with high performance, high availability, and high scalability.

Previous works in big data integration have attempted to move all computing tasks to the cloud, as an efficient way to integrate industrial big data due to the super-computing power of the cloud [17]. However, the bottleneck of big data integration emerges in data transmission [30]. At present, most information saved in companies is in the form of unstructured models; for example, processing requirements are saved as documents. The retrieval and extraction of this information are essential tasks in industrial data analysis. Text mining and natural language processing are two techniques used for knowledge discovery from textual context in documents. Moreover, machine vision algorithms can be applied for the information extraction of graphic data, such as product inspection images, computer-aided design drawings, and so forth. In general, unstructured files take up more storage space than structured data, and are generated at high speed. With traditional integration approaches (e.g., the semantic integration approach proposed by Liu et al. [31]), all unstructured and structured data are uploaded into the cloud application servers for processing. This results in a heavy network traffic load that will influence the everyday operation of manufacturing systems.

Since a heavy network traffic load is raised by frequent big data transmutation in the cloud application [16], there has been a need to extend the cloud-computing framework to the edge of networks, in an approach called fog computing, also known as edge computing or cloudlet computing [32]. Fog computing moves computing tasks to edge devices [33] such as smart phones, wearable devices, and game controllers. With the fast-growing development of sensors and chip technology, some pilot applications have emerged to evaluate the effectiveness of fog computing. Zhang et al. [34] investigated a new computing framework for big data sharing and processing in a collaborative edge environment. The designed framework was able to improve the response latency by processing data at edge devices close to the data sources in order to reduce data transmission to the cloud. Tang et al. [35] presented hierarchical distributed fog computing in smart cities to support the integration of massive data generated by infrastructure components. With the fog-computing technology, anomalous and hazardous events in cities could be identified and responded to in different time latencies. Compared with cloud computing, fog computing has a more flexible architecture and a higher speed response, as has been demonstrated recently in many works [36–38]. Although some applications have already been made,

fog computing is restrictive in many domains due to the limited computing ability of edge devices. In manufacturing systems, industrial big data is collected, stored, and managed by the various industrial servers of an enterprise's information system; the data is then uploaded to the remote cloud center for analyzing. For this reason, industrial data is referred to as "server-attached." These industrial servers have much higher computing abilities than other commercial edge devices (e.g. Android devices), making it possible to deploy simple data-processing tasks to the edge nodes in manufacturing systems.

Inspired by previous fog-computing studies, we herein propose a solution for IBDIS in manufacturing systems in order to address the aforementioned network traffic load and privacy issues. Using fog-computing technology, we designed the Fog-IBDIS paradigm, which moves the algorithms and models to the data generator instead of transmitting the data to the cloud for analysis. In this paradigm, the data is processed in the edge servers, which can extract data directly from the data sources to support data analysis and optimization in low-time latency. In this way, Fog-IBDIS shares only the extracted knowledge with the data customer, and thereby protects data privacy by helping to prevent raw data leakage, since the data customer can hardly infer the raw data from the processed data. In addition, the output volume after data processing is at least two to three orders of magnitude smaller than that of raw data, which eases the network traffic load. For example, only 1 TB of processed data is obtained after correlative analysis with 1 PB of raw data. Thus, Fog-IBDIS uses fog computing to move big data analysis to the data source in order to address the problems caused by cloud computing in IBDIS and improve the data integration efficiency.

3. Fog-IBDIS framework in manufacturing systems

IBDIS manages all the industrial big data emerging from the product life-cycle, including product design data, manufacturing data, marketing data, and so forth. Based on the data source, industrial big data can be divided into two types: system data and IoT data, as shown in Fig. 1. System data refers to data generated from various types of enterprise information systems, such as e-commerce platforms (EPs), social networking platforms (SNPs), product lifecycle management (PLM), enterprise resource planning (ERP), maintenance repair and overhaul (MRO), and supply-chain management (SCM). These information systems accumulate massive product research and development (R&D) data, manufacturing data, supply-chain data, sales data, customer feedback, and more. IoT data refers to data captured by sensors, such as radio frequency identification (RFID) readers and barcode readers. Through sensors such as these that are embedded in intelligent equipment, an enormous amount of production-process data can be automatically collected regarding the equipment in the workshop and the state of the products.

Given that industrial big data is acquired from multiple sources, data integration can be classified into three types: single-source IBDIS (S-IBDIS), cooperative IBDIS (C-IBDIS), and multisystem IBDIS (M-IBDIS). In S-IBDIS, all the data for analysis comes from a single data source, such as the MES. In this mode, the data can be easily integrated through the simple Fog-IBDIS framework. In the C-IBDIS mode, data is acquired from more than two data sources in a manufacturing system. The raw data can be aggregated using data synchronization/asynchronous replication, data federation, or interface-oriented, through middleware, virtual databases, and data warehouse technology. In this mode, raw data can be aggregated directly, since the data exchange is secure within the private industrial Internet of the manufacturing system. M-IBDIS involves data integration from data sources in different manufacturing systems, and is suitable for data analysis in the supply chain or for a multi-plant company. In this mode, raw data is not allowed to be transmitted between manufacturing systems, in order to protect data privacy. Therefore, the M-IBDIS model applies the analytical algorithms to the data sources individually, aggregates the intermediate results, and uploads the final results to the data customers.

In Fog-IBDIS, the integration process is defined as an IBDIS task, which is abstracted as a TFG. The TFG is a graph that represents an IBDIS task described by a number of task nodes including dataset nodes, operation nodes, and transmission nodes. The dataset node refers to the virtual data needed in the data processing; it provides the dataset name, data structure, and feed path. The dataset node provides a view of the integrated data for data consumers, which contains the data structure, feed path, and some interpretative samples. With the data view, a potential data customer can understand the data easily. The operation node is a unit for data processing that defines the input, the output, and the specific algorithms used in the data analyzing. Industrial big data analyses contain several kinds of operation nodes, such as the data-cleaning node, the data transformation node, the prediction node, and the clustering

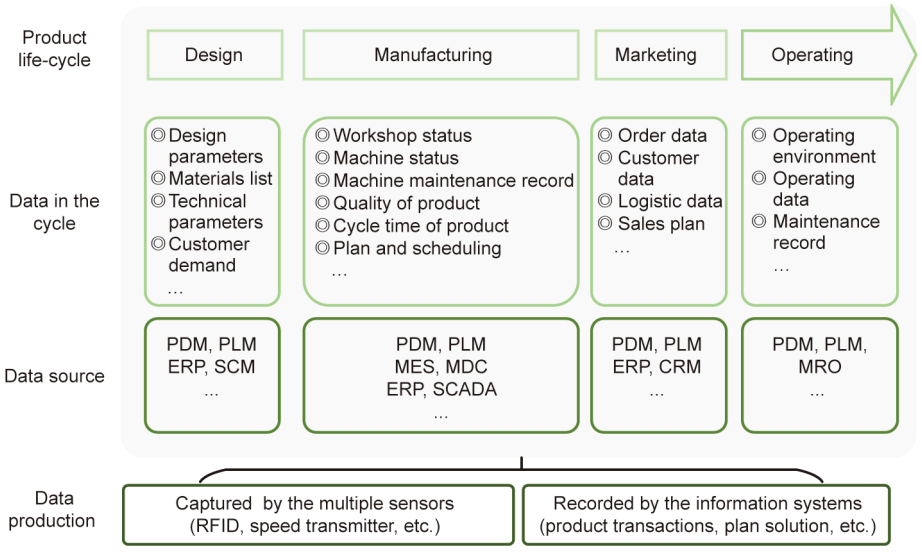

Fig. 1. Industrial big data for the product life-cycle. CRM: customer relationship management; ERP: enterprise resource planning; MDC: manufacturing data collecting; MRO: maintenance repair and overhaul; PLM: product lifecycle management; RFID: radio frequency identification; SCM: supply-chain management.

node. The data transmission node transfers the intermediate results between two manufacturing systems in order to relay the operation process. To illustrate how to make use of the TFG, a sample TFG is shown in Fig. 2 containing three IBDIS tasks. Task 1 contains a dataset node (D) and operation node (O), which is an S-IBDIS task processing data from a single data source, D1. The output of the operation node O1 in Task 1 is sent to the operation node O2, which belongs to Task 2. Task 2 is a C-IBDIS processing data from two data sources (D1 and D2) in the same manufacturing system. The output of operation node O2 is uploaded by Fog-IBDIS to Task 3, which is an M-IBDIS task finished in another manufacturing system. Hence, the output of Task 2 is transmitted by the transmission node T1 to Task 3. With the data provided by T1 and D3, three operation nodes are developed to finish Task 3. With the TFG, the integration process can be clearly described through the three kinds of nodes.

In Fog-IBDIS, all tasks are finished by the data owner, and only the analysis result is uploaded to the data customer; this protects data privacy and eases the network traffic load. To finish the IBDIS task, Fog-IBDIS contains one fog server and several fog clients on the hardware. The fog server helps to control the IBDIS process, and clients running on the edge nodes of the network execute the tasks defined in the TFG. As shown in Fig. 3, the fog server first issues the operation task, Task 1, to the fog client through the operation instruction I1. This task contains two operation nodes, O1 and O2, which process datasets from two data sources (output of D1 and D2). Next, the processed result is transmitted to the subsequent fog client through the data flow DF4–T1–DF5. The fog client in data generator 2 analyzes the data from T1 and D3 according to Task 2, and subsequently uploads the result to the fog server through the data flow DF6. The fog server then transmits the analysis result to the data customer. During the analysis, all the raw data-processing tasks are finished in the fog client, which is implemented in the data generator company, in order to protect data privacy. Only the intermediate results are transferred between the fog clients, in order to ease the network traffic load.

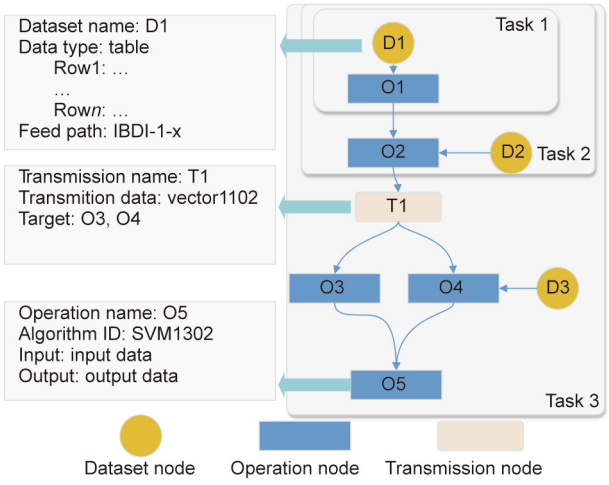

Fig. 2. A TFG for data integration with Fog-IBDIS. IBDI: industrial big data integration.

4. The functional modules of Fog-IBDIS

To meet the requirements of fog-computing-based big data integration, the functional modules (Fig. 4) of Fog-IBDIS are presented in this section, and include: TFG management, compilation and running control, the data integration model, the basic algorithm library, and the management component.

4.1. TFG management

TFG management is the core function of Fog-IBDIS, as it edits the flow graph of the integration task. This module contains four parts: dataset node management (DNM), operation node management (ONM), transmission node management (TNM), and edge management (EM). DNM maintains the dataset node in the TFG, which consists of data structure definition, data formatting, and feed path definition. Through the DNM module, the data generator can edit and release the data through a data view, which describes the information of a dataset in detail. This allows the data to be easily accessible through the Fog-IBDIS platform; the data customer can search and take the data for analysis through the provided data view. The ONM provides the editing, packaging, and checking functions for operations in IBDIS, such as data cleaning, data transformation, analyzing, and so forth. Through the ONM module, all the operations for an analytical case are packaged into several nodes by the data customer. Furthermore, all operation nodes are checked by the data generators to ensure raw data security. The TNM manages the transmission process in the data analysis, including data encoding, decoding, uploading, and offloading. Since all computing tasks are finished by the data generator, a large analytical case is usually distributed after being finished by several data generators. To increase transmission security, the data is encoded as cipher text with private keys during transmission. The EM module defines and checks the data structure that is automatically transferred during two adjacent nodes to ensure the validity of the TFG.

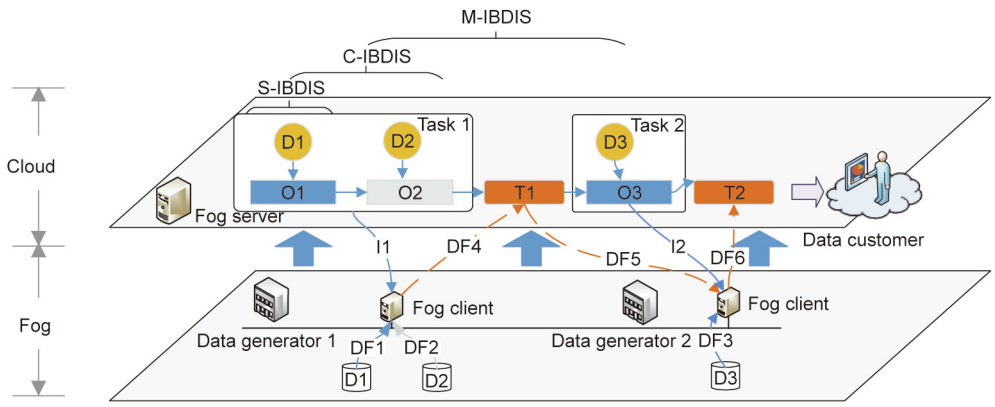

Fig. 3. Schematic diagram of Fog-IBDIS. DF: data flow; I: instruction.

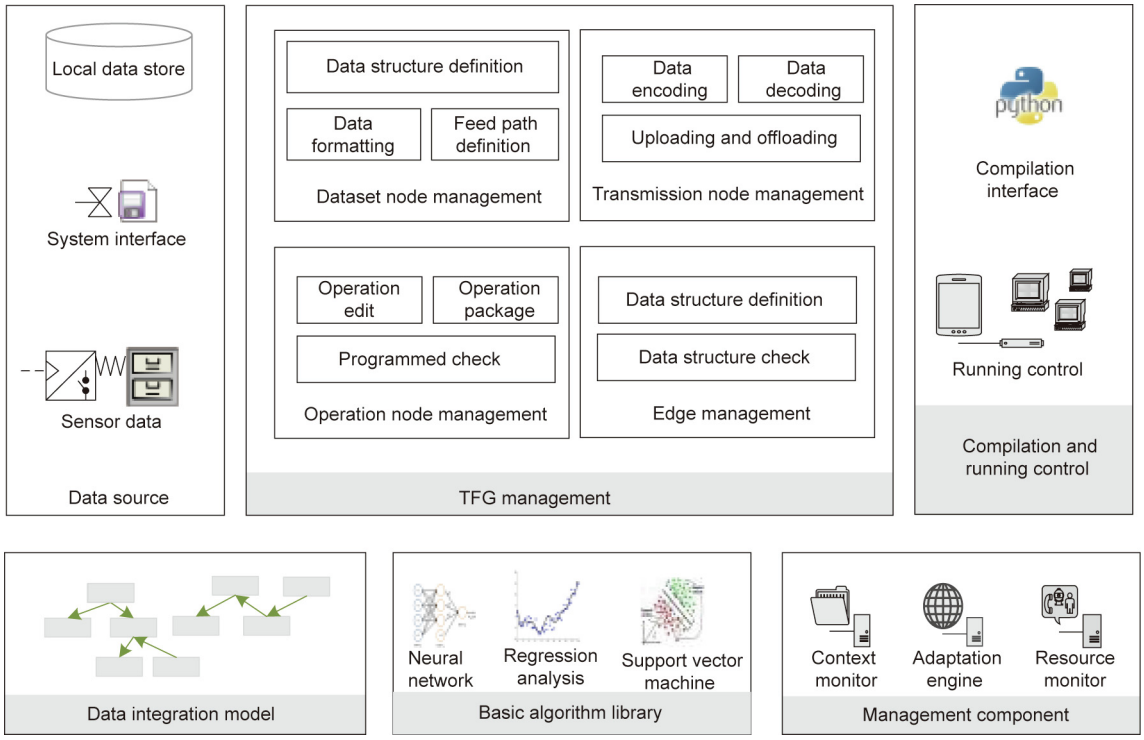

Fig. 4. The functional modules of the Fog-IBDIS platform.

4.2. Compilation and running control

The Fog-IBDIS platform provides a python application programming interface for compilation and running. All the debug tasks are finished by the data customer with the sample datasets. After the debugging, the IBDIS tasks are compiled by the python compiler and sent to the data generator through the Fog-IBDIS platform. The compiled files are distributed within several Fog-IBDIS servers belong to different manufacturing systems, based on the TFG. The running environment and configuration parameters are determined and adjusted by the Fog-IBDIS platform.

4.3. Data integration model

The big data integration model defines the transformation of the data to get it ready for loading into the operation node. The data items defined in the operation are diverse in terms of data source and structure. To convert the data for operation, the data integration model has four layers of components: target data items, entities, systems, and source data items (Fig. 5).

- **Target data item.** This contains a collection of data items to support the specific big data analysis case in the data model.
- **Entity.** This is used to identify the target data items. All the data items belonging to an entity are one-to-one correspondent with the entity ID. The data items belong to different entities are correlated with each other through the entity relationship chain.
- **System.** This defines the access path of the data item, including the systems managing the data items, the data integration mode, and so forth. With the system component, the data item can easily be extracted through the different interfaces from the information systems.
- **Source data item.** This defines the source field in the database and interface. With the source data item, the data integration unit can accurately pinpoint the location to extract the raw data.

If the source data items belong to two different domains, the entity relationship chain should be defined to specify the relationship of these source data items. For example, the data items named "quality" and "cycle time" are both the properties of the product entity. As another example, the data item named "OEE" belongs to the "machine" and the data item name "quality" belongs to the "product." In the data analysis scenario, the two data items are correlated together with the entity relationship chain "product process route–machine."

4.4. Basic algorithm library

The basic algorithm library provides the underlying support for simple data analysis in the data transformation in Fog-IBDIS. Data customers can invoke and modify the basic algorithm to meet the requirements of data processing such as outlier analysis, missing value interpolation, and other functional requirements. Basic machine learning methods such as neural networks, regression analysis, support vector machines, and other advanced machine learning algorithms can be applied in data analysis with the module.

4.5. Management component

The management component ensures that the far-edge computing architecture can run effectively in different kinds of environments. In general, a complete management component contains three main parts: the context monitor, the resource monitor, and the adaptation engine. The context monitor and resource monitor are used to monitor the status information of the devices, including resource availability and real-time task-executing information. The adaptation engine can realize data reduction and system compatibility.

5. Case study

To illustrate the effectiveness of Fog-IBDIS, we present a case study involving integration of the big data of a commercial

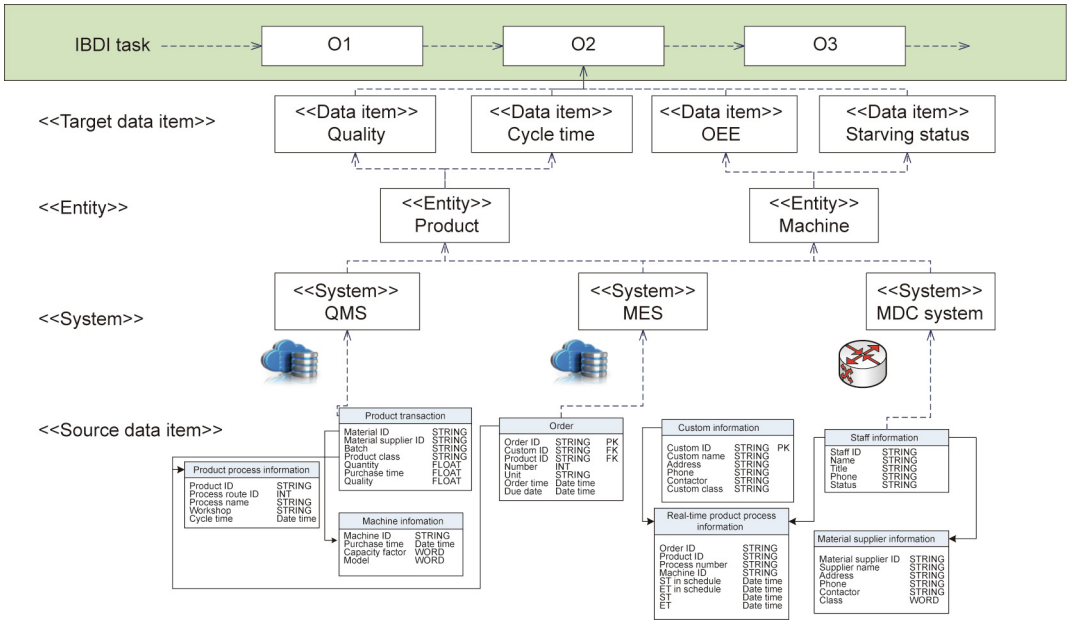

Fig. 5. Data integration model for IBDIS with fog computing. OEE: overall equipment effectiveness; QMS: quality management system.

aircraft-manufacturing group in Shanghai. In this company, one type of plane is assembled with a sub-assembly manufacturing system and is designed by the advanced R&D center, which belongs to two individual companies. During the plane assembly, different processes have different impacts on the stress of the positioner, which is a key piece of equipment for fixing and supporting the plane. The status of a plane can be estimated through an analysis of the positioner status. In this case, the advanced R&D center wants to optimize the process route by monitoring the pose deformation of the wing with the assembly processing. This case requires the pose data (including the position of several key points on the wing) and the process transaction data (containing the execution time of each process step). The pose data can be exported from the manufacturing data collecting (MDC) system belonging to the sub-assembly manufacturing system, and is the data generator. The process transaction data belongs to the MES managed by the planning and scheduling group in the advanced R&D center, and plays the role of both data generator and data customer in this case.

5.1. Fog-IBDIS-based data integration for process route optimization

To integrate the data for the process route optimization, a TFG with two tasks is designed with two dataset nodes, three operation nodes, and one transmission node (Fig. 6). The first operation node preprocesses the position of several key points of the wing, including data cleaning and transformation. The second operation node detects abnormal pose deformation by the position analysis of the key points of the wing. The abnormal data fragments are then transferred to the third operation node in Task 2 through the Fog-IBDIS platform. In this operation node, these data fragments are mapped to the process transaction data in order to diagnose the root assembly process causing the abnormal pose deformation.

The first operation node O1 preprocesses the pose status of the wing, which is measured by the force feature of the three joint points shown in Fig. 7(a). The force data is sensed by the sensors embedded in the positioner, and is collected by the SCADA system through open platform communications technology. First, the null value and abnormal values in the data records are detected and restored in the data cleaning, as shown in Fig. 7(b). Next, according to the data integration model, the data items are transformed through the data cube, which is designed to customize the raw data into the structure of the target data (data transformation in business) with operations such as drill-down, roll-up, slice, dice, and pivot. For example, we have a dataset containing transaction data about the status of a positioner. The dataset contains records in three dimensions: machine, field, and time. Each cell (M, F, T) of the cube contains the value of the field F of the machine M at the time T. In this example, the z-axis force of the positioners LWA, LWF, and LWO at time $T4$ is needed in the big data analysis. The operation "dice–slice–slice–roll-up" is designed to obtain the target data in this case, as shown in Fig. 7(c).

The second operation node O2 detects the abnormal pose deformation of the wing and sends the abnormal data fragments to Task 2. During the data analysis of the position of the three key points, the deviation of the wing's setting angle (described by the E1, E2, and E3), the deviation of the positive-dihedral angle (described by E4 and E5), and the deviation of the sweep angle (described by E6) are estimated using the status analysis model (Fig. 7(a)). The abnormal deviation of the wing status is detected by a control chart for detecting abnormal situations. Data points are regarded as anomalous when errors exceed the upper and lower control limits (e.g., error within 5%). The unusual data are detected and transferred to Task 2 through Fog-IBDIS. The deviation data is then mapped to the third operation node "O3," which diagnoses the root cause of the unusual deviation through a combined analysis of the deviation data and the assembling process by special experts. As a result, recommendations for improvement are obtained in order to improve the assembly process and reduce the degree of pose deformation.

Within this case, Fog-IBDIS manages the industrial big data emerging from the aircraft-manufacturing system and distributes the data integration tasks to the fog clients to provide data for analysis to optimize the process route. As can be seen in Fig. 7(d), the wing posture has undergone great changes from September 10 to September 14. It is inferred from the data analysis that this change is caused by irrationality of the process route: a fixture connection was removed before the deburring cleaning of the joint hole in the left wing of the plane. In this case, the volume of the collected raw data from August 1 to September 30 in 2016 is 1.7 TB, whereas the volume of the transmitted intermediate results is 160 kB. In the

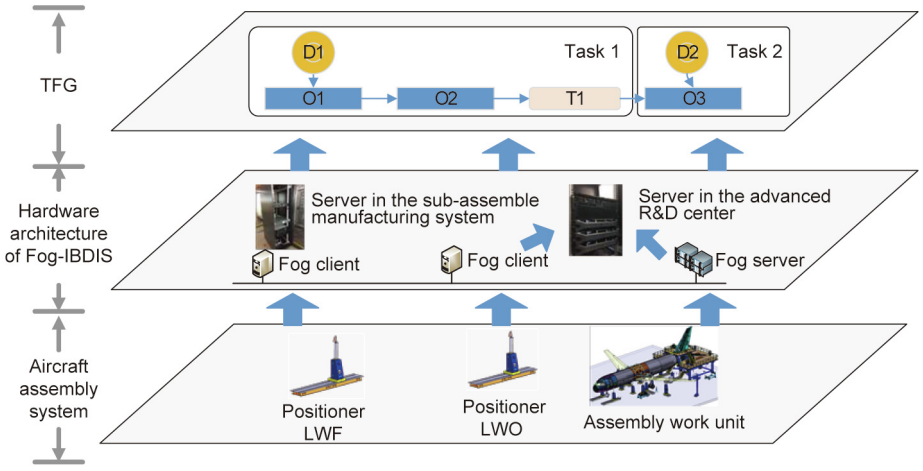

Fig. 6. The Fog-IBDIS task for process route optimization. The LWF and LWO are the positioners for the left wing.

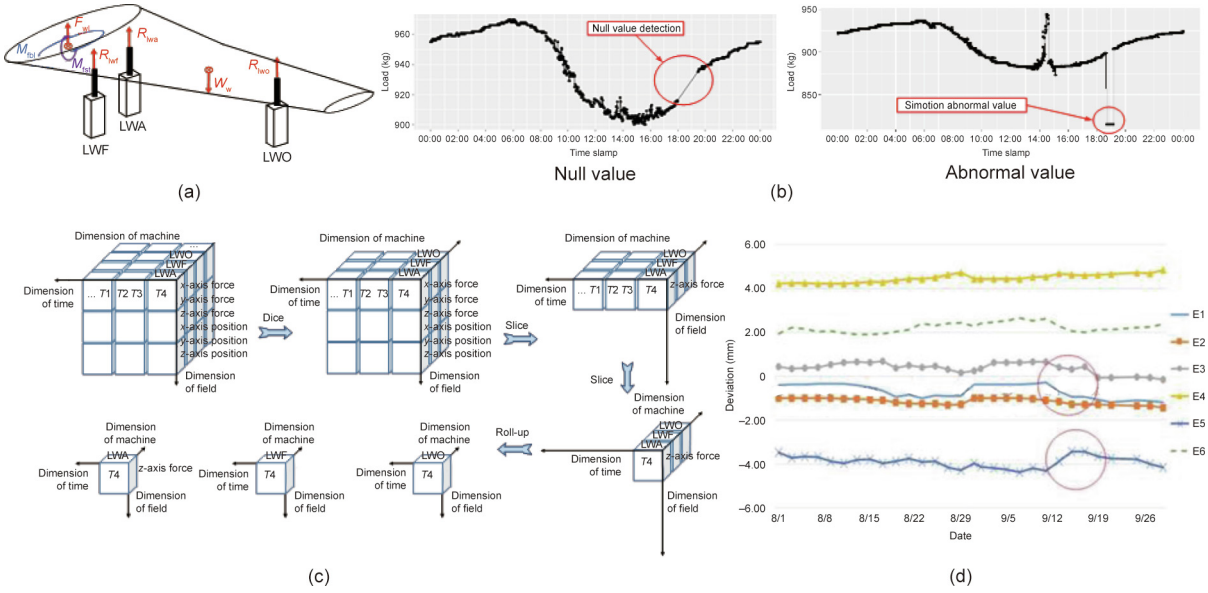

Fig. 7. Positioner status monitoring and analysis through Fog-IBDIS. (a) Status analysis model; (b) data cleaning; (c) data cube model for data preprocessing; (d) the analyzed deviation of the three angles. F_{wl}: the bearing reaction from the body to the left wing; M_{fbl}: the moment of force around the transverse; M_{fst}: the moment around the spanwise; R_{lwf}: the bearing reaction from the positioner LWF to the left wing; R_{lwa}: the bearing reaction from the positioner LWA to the left wing; R_{lwo}: the bearing reaction from the positioner LWO to the left wing; W_w: the gravity of the left wing; E1, E2, E3: deviation of the wing's setting angle; E4, E5: the deviation of the positive-dihedral angle; E6: the deviation of the sweep angle.

data transmission of this case study, the volume of the transmitted data is only 9.1×10^{-8} of the original data volume, which indicates that Fog-IBDIS can greatly reduce the volume of transmitted data to ease the network traffic load.

5.2. Implementation of Fog-IBDIS

In this case, Fog-IBDIS is implemented to integrate industrial big data. The implementation process (Fig. 8) is divided into five steps, as follows:

Step 1: model design. In the Fog-IBDIS, the big data integration model is first described to define the data structure for the transformation. With the data model, all data items for the same analysis subject are linked together with the semantic triples, and the relationships between the target data items and the raw data items are defined.

Step 2: TFG design. According to the big data integration model, all nodes are defined in the TFG to finish the big data integration and sharing. These nodes are packaged into two tasks, and each task is programmed with python by the data customer.

Step 3: task debugging. The tasks are compiled and debugged according to the specific TFG by the data customer with the sample data instances contained in the data view. The compiled applications are launched in the fog clients for testing and debugging in order to verify the design of the TFG and codes through the Fog-IBDIS platform.

Step 4: task review. After program debugging and revision, the tasks are submitted to the data generator to obtain permission for task operation.

Step 5: task execution. After the task review, the tasks are implemented in the fog clients coordinated by the Fog-IBDIS platform in order to integrate the industrial big data.

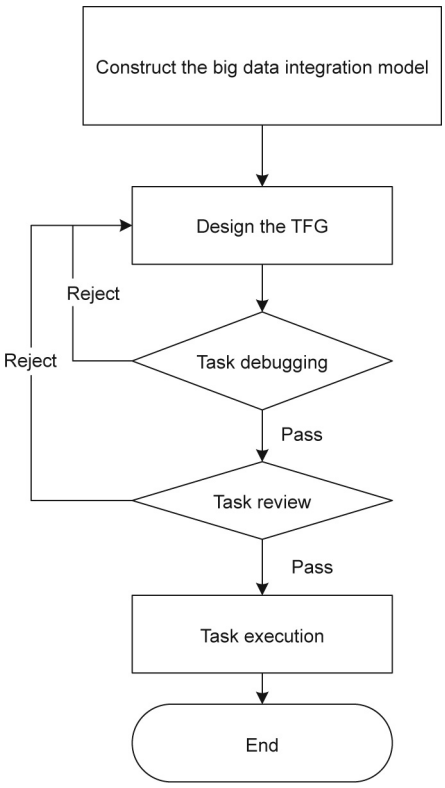

Fig. 8. The implementation process of Fog-IBDIS.

5.3. Discussion

To further demonstrate the effectiveness of Fog-IBDIS, a careful comparison of Fog-IBDIS and IBDIS with cloud computing is provided. Fog-IBDIS differs greatly from IBDIS with cloud computing in terms of the data integration framework and integration performance (time latency, data privacy, and network traffic load).

Decentralization. In IBDIS with cloud computing, the data generated by IoT devices and information systems wait for passive extraction by the big data center, and all data extraction, transformation, and loading tasks are completed by the centralized big data platform. Although the big data center has immense computing power, it is quite challenging to segregate, index, store, and clean a large amount of industrial big data. With fog computing, Fog-IBDIS decentralizes the massive data-processing tasks of an analytic case into the edge nodes, which effectively relieves the pressure of a big data platform in manufacturing systems.

Latency. In contrast to cloud computing, fog computing parallelizes data processing at the edge of the network, which satisfies the requirements of real-time data analysis for industrial control. With Fog-IBDIS, industrial big data is processed at the generative side, then transmitted into the cloud side to enable further analysis; this supports data analysis at different levels of time latency.

Data privacy. In a data-sharing service with cloud computing, the shared data can easily be copied, shared, and secondary traded. Original data privacy cannot be guaranteed, since the service shares original data directly. In industry, original data is usually strictly confidential, and is only allowed to be used within manufacturing systems. Unlike the cloud-computing-based IBDIS, which transfers all raw data to the cloud data center, Fog-IBDIS only transmits the intermediate results to the next fog client and uploads the analytical results to the data customer. In addition, Fog-IBDIS introduces the task review mechanism, which allows the data generator to check the raw data security; this protects the data privacy in the data integration.

Network traffic load. In cloud computing, all raw data are transmitted to the big data center, which is equipped with a distributed storage platform such as a Hadoop distributed file system. Some of the raw data in manufacturing systems is very large in size, such as massive images scanned by laser scanning, and a great deal of bandwidth is required to upload all these images. With Fog-IBDIS, only the intermediate processed results are selected and uploaded through the network. The size of most of the raw data is reduced through data cleaning, resampling, and transformation, so the cost of network communication is effectively reduced. Moreover, unlike the batch processing of cloud computing, the decentralization that occurs as Fog-IBDIS uploads data to different data points reduces the peak load of the networks.

6. Conclusions

This paper investigates Fog-IBDIS, a big data integration and sharing framework with fog computing, from three aspects: its operation principle, functional modules, and implementation. Unlike previous big data integration studies with cloud computing, this study constructs the Fog-IBDIS platform using fog computing, which splits the IBDIS task into several sub-tasks run by the data generators. Regarding data processing, all raw datasets are preprocessed and analyzed by the data owners to protect the raw data privacy. In addition, Fog-IBDIS applies the data processing in the fog clients within the manufacturing systems, thereby changing the centralized data-processing mode into distributed task execution. Accordingly, only the analyzed results are transferred between the distributed fog clients, which reduces the volume of transmitted data and eases the network traffic load. To the best of our knowledge, this is the first attempt to combine IBDIS with fog-computing technology in manufacturing systems.

In future research, we will address industrial big data analysis with fog computing and the control of edge devices in manufacturing systems.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (51435009), Shanghai Sailing Program (19YF1401500), and the Fundamental Research Funds for the Central Universities (2232019D3-34).

Compliance with ethics guidelines

Junliang Wang, Peng Zheng, Youlong Lv, Jingsong Bao, and Jie Zhang declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Hughes D, Ueyama J, Mendiondo E, Matthys N, Horré W, Michiels S, et al. A middleware platform to support river monitoring using wireless sensor networks. J Braz Comput Soc 2011;17(2):85–102.
[2] Jiang P, Ding K, Leng J. Towards a cyber–physical–social-connected and service-oriented manufacturing paradigm: social manufacturing. Manuf Lett 2016;7:15–21.
[3] Wang JL, Zhang J. Big data analytics for forecasting cycle time in semiconductor wafer fabrication system. Int J Prod Res 2016;54(23):7231–44.
[4] Wang JL, Zhang J, Wang XX. Bilateral LSTM: a two-dimensional long short-term memory model with multiply memory units for short-term cycle time forecasting in re-entrant manufacturing systems. IEEE Trans Industr Inform 2018;14(2):748–58.
[5] Zhang WJ, Lin Y. On the principle of design of resilient systems—application to enterprise information systems. Enterprise Inf Syst 2010;4(2):99–110.
[6] Zhang WJ, van Luttervelt CA. Toward a resilient manufacturing system. CIRP Ann 2011;60(1):469–72.
[7] Tsuda T, Inoue S, Kayahara A, Imai S, Tanaka T, Sato N, et al. Advanced semiconductor manufacturing using big data. IEEE Trans Semicond Manuf 2015;28(3):229–35.

[8] Lu C, Li X, Gao L, Liao W, Yi J. An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times. Comput Ind Eng 2017;104:156–74.
[9] Lei CU, Man KL, Liang HN, Lim EG, Wan KY. Building an intelligent laboratory environment via a cyber–physical system. Int J Distrib Sens Netw 2013;9 (12):109014.
[10] Wang JL, Zhang J, Wang XX. A data driven cycle time prediction with feature selection in a semiconductor wafer fabrication system. IEEE Trans Semicond Manuf 2018;31(1):173–82.
[11] Wang W, Chong W, Liu D, Liang HN, Man KL, Han YS, et al. An examination of the internet of things through the data management perspective. J Platf Technol 2014;2(2):16–30.
[12] Lu C, Gao L, Li XY, Chen P. Energy-efficient multi-pass turning operation using multi-objective backtracking search algorithm. J Clean Prod 2016; 137:1516–31.
[13] Lu C, Gao L, Li XY, Xiao SQ. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. Eng Appl Artif Intell 2017;57:61–79.
[14] Kusiak A, Xu GL. Modeling and optimization of HVAC systems using a dynamic neural network. Energy 2012;42(1):241–50.
[15] Zhong RY, Xu C, Chen C, Huang GQ. Big data analytics for physical Internet-based intelligent manufacturing shop floors. Int J Prod Res 2017;55(9):2610–21.
[16] Zhang W. An integrated environment for CAD/CAM of mechanical systems [dissertation]. Delft: TU Delft; 1994.
[17] Majkić Z. Big data integration theory: theory and methods of database mappings, programming languages, and semantics. Heidelberg: Springer; 2014.
[18] Wang G, Gunasekaran A, Ngai EWT, Papadopoulos T. Big data analytics in logistics and supply chain management: certain investigations for research and applications. Int J Prod Econ 2016;176:98–110.
[19] Mourtzis D, Vlachou E, Milas N. Industrial big data as a result of IoT adoption in manufacturing. Procedia CIRP 2016;55:290–5.
[20] Lim JB, Yu HC, Gil JM. An efficient and energy-aware cloud consolidation algorithm for multimedia big data applications. Symmetry 2017;9(9):184.
[21] Kadadi A, Agrawal R, Nyamful C, Atiq R. Challenges of data integration and interoperability in big data. In: Proceedings of 2014 IEEE International Conference on Big Data; 2014 Sep 27–30; Washington, DC, USA. Piscataway: IEEE; 2015. p. 38–40.
[22] Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Khan SU. The rise of "big data" on cloud computing: review and open research issues. Inf Syst 2015;47:98–115.
[23] Wang JL, Yang JG, Zhang J, Wang XX, Zhang WJ. Big data driven cycle time parallel prediction for production planning in wafer manufacturing. Enterprise Inf Syst 2018;12(6):714–32.
[24] Pan WK, Yang Q, Aggarwal C, Koch C. Big data. IEEE Intell Syst 2017;32(2):7–8.
[25] Xiang F, Yin Q, Wang Z, Jiang GZ. Systematic method for big manufacturing data integration and sharing. Int J Adv Manuf Technol 2018;94(9–12):3345–58.
[26] Ma'ayan A, Rouillard AD, Clark NR, Wang ZC, Duan QN, Kou Y. Lean big data integration in systems biology and systems pharmacology. Trends Pharmacol Sci 2014;35(9):450–60.
[27] Mezghani E, Exposito E, Drira K, Da Silveira M, Pruski C. A semantic big data platform for integrating heterogeneous wearable data in healthcare. J Med Syst 2015;39(12):185.
[28] Jiang L, Xu LD, Cai H, Jiang Z, Bu F, Xu B. An IoT-oriented data storage framework in cloud computing platform. IEEE Trans Industr Inform 2014;10 (2):1443–51.
[29] Chang BR, Tsai HF, Tsai YC, Kuo CF, Chen CC. Integration and optimization of multiple big data processing platforms. Eng Comput 2016;33(6):1680–704.
[30] Suárez-Albela M, Fernández-Caramés TM, Fraga-Lamas P, Castedo L. A practical evaluation of a high-security energy-efficient gateway for IoT fog computing applications. Sensors 2017;17(9):E1978.
[31] Liu X, Zhang WJ, Radhakrishnan R, Tu YL. Manufacturing perspective of enterprise application integration: the state of the art review. Int J Prod Res 2008;46(16):4567–96.
[32] Varghese B, Wang N, Barbhuiya S, Kilpatrick P, Nikolopoulos DS. Challenges and opportunities in edge computing. In: Proceedings of IEEE International Conference on Smart Cloud; 2016 Nov 18–20; New York, NY, USA. Piscataway: IEEE; 2016. p. 20–6.
[33] Shi W, Dustdar S. The promise of edge computing. Computer 2016;49 (5):78–81.
[34] Zhang Q, Zhang XH, Zhang QY, Shi WS, Zhong H. Firework: big data sharing and processing in collaborative edge environment. In: Proceedings of the 4th IEEE Workshop on Hot Topics in Web Systems and Technologies; 2016 Oct 24–25; Washington, DC, USA. Piscataway: IEEE; 2016. p. 20–55.
[35] Tang B, Chen Z, Hefferman G, Wei T, He H, Yang Q. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In: Proceedings of the ASE Big Data & Social Informatics; 2015 Oct 7–9; Kaohsiung, Taiwan, China. New York: ACM; 2015.
[36] Kumar N, Zeadally S, Rodrigues JJPC. Vehicular delay-tolerant networks for smart grid data management using mobile edge computing. IEEE Commun Mag 2016;54(10):60–6.
[37] Liu JQ, Wan JF, Zeng B, Wang QR, Song HB, Qiu MK. A scalable and quick-response software defined vehicular network assisted by mobile edge computing. IEEE Commun Mag 2017;55(7):94–100.
[38] Park HD, Min OG, Lee YJ. Scalable architecture for an automated surveillance system using edge computing. J Supercomput 2017;73(3):926–39.