Edge Computing-Enabled Resource Provisioning for Video Surveillance in Internet of Vehicles

Xiaolong Xu[1,2], Qi Wu[1], Chengxun He[1], Shaohua Wan[3], Lianyong Qi[4(✉)], and Hao Wang[5]

[1] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing, China

[3] School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, China

[4] School of Information Science and Engineering, Qufu Normal University, China
lianyongqi@gmail.com

[5] Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjϕvik, Norway

Abstract. As a novel technology, Internet of Vehicles (IoV) is employed to gather real-time traffic information for drivers from sensors and video surveillance devices with image processing, circumstances analysis and events recognition. In spite of multiple advantages of IoV, preprocessing the huge data may demand abundant computation resources for video surveillance devices. Migrating tasks to remote servers for performing is efficient to solve this problem, but it needs high network bandwidth, which causes traffic congestion and delay. Edge computing has capability to enhance processing performance, which complements video surveillance device and addresses numerous shortcomings. Nevertheless, edge computing for video surveillance remains a challenge to achieve low-latency and load balance through limited amount of edge servers. To handle this challenge, an Edge computing-enabled Resource Provisioning Method (ERPM) for Video Surveillance in IoV is proposed in this paper. Technically, SPEA2 (improving the Strength Pare to Evolutionary Algorithm) is picked to solve the multi-objective optimization problem aiming at minimizing the time consumption and optimizing load balance. Finally, experimental simulation for Evolution algorithm demonstrate the appropriation and efficiency of ERPM.

Keywords: Internet of Vehicles; Edge Computing; Resource Allocation; Video Surveillance.

1 Introduction

From the past few years, with the development of communication and networking technologies, e.g., wireless sensor networks, 5G communication networks and short-range wireless communication, the Internet of Things (IoT) is evolving to a paradigm that achieves special prominence to the interconnection of physical objects and human inhabitant [1] [2]. As an emerging technology of IoT, Internet of Vehicles (IoV) consists of vehicles, intelligent mobile cameras, sensors, actuators and applications through internet, which has significant affects in smart city [3]. IoT technology enables devices to collect and analyze the traffic information that comprises position, driving status and road condition from the sensors of the vehicles and video surveillance devices. The majority of traffic information about traffic condition is gathered from video surveillance devices, and preprocessed in video surveillance systems, which supports for image processing, traffic condition analysis and events recognition. Then, video surveillance systems have crucial impact in IoV.

Meanwhile, the proliferation of video surveillance equipment has been a fundamental device deployed in public places, such as shopping centers, streets, schools, city facilities, and home. In 2012, approximately 8 million surveillance devices worldwide were connected to the Internet and the number is expected to grow to 170 million in 2021 [4]. To preprocess the massive amounts of all-day-operating video data at the camera nodes (e.g., extracting features), video surveillance devices demands abundant local computation resources. Whereas in reality, these devices cannot sustain the huge work, and then migrate the tasks to the remote servers to release the pressure, which the migration needs high network bandwidth and lead to significant delay [5]. Obviously, the paradigms cannot satisfy the requirement of real-time video processing and analyzing tasks.

Recently, as an emerging technique of distributed computing for the preprocessing video data, edge computing enables video surveillance devices to reduce the transmission delay [6]. Video surveillance devices deploy edge node with computing, storage, and network connectivity through switches, routers, embedded equipment, electronic facilities. By supporting the edge computing, the communication and computation capabilities of video surveillance systems can be utilized to deal with the computing tasks that include video compressing, preprocessing and analyzing through short-range wireless communications [7]. However, the transmission of a mass volume of video data may lead to congestions and delays due to limited network bandwidth. In this paper, in order to improve the processing capability of monitoring terminals, we present a brand new smart resource allocation for video surveillance system in edge computing. Taking traffic monitoring as an instance of work, the proposed edge computing based surveillance system is able to obtain the video data information for traffic condition in real-time, and offload the real-time data to cloud computing center [8]. The captured data been analyzed and filtered by edge servers firstly that mitigate simultaneously the workload of the transmit network and the cloud data center [9] [10].

The main contributions of this paper are summarized as:

-Analyze the details of process of tasks computing for Video Surveillance in Internet of Vehicles with Edge Computing enabled.

-Formulate the resource provisioning problem as a multi-objective optimization problem to optimize the time consumption and load balance of ENs (Edge Nodes).

-Adopt an method named ERPM by employing SPEA2 (Improving the strength pare to evolutionary algorithm) with simulation experimental to solve the objective problem we proposed.

The remainder of this paper is organized as follows. In the next Section, system model and problem formulation is discussed. Furthermore, an edge computing-enabled resource provisioning method for video surveillance in internet of vehicles is provisioned in Sections 3. Experimental evaluation is presented in Section 4. After that, we present related work in Section 5. Finally, conclusion and future works are drawn in Section 6.

2 Related work

The technical aspects and potential benefits of edge computing have been researched extensively in the recent literature. The video surveillance system in edge computing can be leveraged to improve the performance of processing tasks from monitoring terminals [11] [12]. An efficient resource-allocation tactics and a real-time video packet scheduler are presented in [13] that formulates an effective real-time video uplink framework to enhance utility, boost transfer efficiency, and stabilize image quality. Reference [14] employs Deep Learning (DL) algorithms to study a Distributed Intelligent Video Surveillance (DIVS) system which migrates computing workloads from the network center to edges to reduce tremendous network communication overheads and provide low-latency and precise video analysis solutions. In [15], Puvvadi et.al. explored a new protocol to drastically weaken the delay to execute computing tasks (e.g., cryptographic mechanisms) and increase the supported bit rate compared with the baseline, while providing desirable security features.

The smart edge computing for video surveillance devices is proposed to provision the computation and allocate resources for the execution of computation tasks. However, when employing the edge servers to accommodate the computing tasks, the scarce computing capability of edge servers should be prioritized. In other words, the supplies of joint running computing tasks on edge server must be prevented. In this situation, the computing tasks should be properly distributed to various edge servers for execution. The trade-off between limited bandwidth and resource allocation in edge computing framework is investigated in [4] where the authors exploit an framework for cooperative video processing to send back a few video features to remote servers through nearby edge nodes while delivering original video would lead to bandwidth starvation.

In [16], X. Xu et.al. emploied Non-dominated Sorting Genetic Algorithm II (NSGA-II) to accomplish multi-objective optimization to shorten the offloading time of the computing tasks and reduce the energy consumption of the edge computing nodes. Different from the existing work (NSGA-II), our design target is to exploit an appropriate and efficient way through SPEA2 to allocate resources via edge nodes, which deploys legitimately edge nodes and ensures high edge resource utilization [17] [18].

3 System Model and Problem Formulation

In this section, we present a system framework for video surveillance at traffic road in edge computing. In the monitoring spot, there is M smart video surveillance equipment, denoted as $R = \{r_1, r_2, \cdots, r_M\}$, along the roadside. The smart video surveillance equipment consist of monitoring terminals, APs and edge servers. In this framework, monitoring terminals transmit video datum through APs. The acquiring video datum accesses edge servers to processing. Edge servers have powerful computing ability that can execute complicated computing tasks, and the cost of servers stays at a relatively high level; thus, it is unreasonable to deploy one edge server on every control rod. Then, there are W edge servers, denoted as $S = \{s_1, s_2, \cdots, s_W\}\,(W < M)$ and each edge server consists of L virtual machine (VM) instances, denoted as $V = \{v_1, v_2, \cdots, v_L\}$. Suppose the capacity of edge server equates to the amount of VM instances in corresponding edge server. Then, the capacity of the $m-$th ($m = \{1, 2, \cdots, M\}$) server s_M can bear the number of VM which is denoted as a_M. Assume $v_{w,l}$ represents the $l-$th ($l = \{1, 2, \cdots, L\}$) VM in the s_w ($w = \{1, 2, \cdots, W\}$). Assume that there are J computing tasks running on the edge servers, denoted as $T = \{t_1, t_2, \cdots, t_J\}$, and $t_{m,j}$ represents the $j-$th computing task in the $m-$th video surveillance equipment.

3.1 Time consumption model

The time consumption consists of four parts, i.e., the time of migration between adjacent access points, the computing time in corresponding edge server, feedback time and the offloading time from edge servers to cloud center. To better support for advanced functions of video surveillance equipment, we aim to minimize the time consumption.

When the video data generated by monitoring terminals without edge server access the nearby edge servers, the process will have the time of migration, which is calculated by

$$g_m^n = \sum_{j=1}^{J} q_m^n \frac{f_{m,j}}{\alpha} \tag{1}$$

where α is the data transmission rate between APs, q_m^n is a binary variable to judge whether the $j-$th computing task in the $m-$th video surveillance equipment is transmitted from s_m to s_n, which is defined by

$$q_m^n = \begin{cases} 1, & t_{m,j} \text{ is transmitted from } s_m \text{ to } s_n, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The computing time of r_m is to be determined by the number of VM instances in edge server, the length of computing task and the processing performance of each VM instances. Accordingly, the task length is denoted as F_m, and the processing performance of each VM is denoted as p. The computing time is calculated by

$$h_m = \sum_{j=1}^{J} \frac{F_m}{a_m \cdot p} z_{m,j} \tag{3}$$

where $z_{m,j}$ judges whether $t_{m,j}$ has been transmitted, i.e. each video surveillance equipment obtains amount of computing tasks that process in the local or nearby edge nodes, which is defined by

$$z_{m,j} = \begin{cases} 0, \; t_{m,j} \text{ has been transmitted,} \\ 1, \; \text{otherwise.} \end{cases} \tag{4}$$

The edge nodes that handle the executing task from nearby nodes should return the results to the original node, which generate the feedback time. The feedback time of r_m is calculated by

$$b_m = \sum_{j=1}^{J} q_m^n \frac{f_{m,back}}{\alpha} \tag{5}$$

where $f_{m,back}$ is the data size of returned results.

The processed video data (e.g. extracting feature) that only have small volume will offloading from the network edge to center. The offloading time of rm is calculated by

$$o_m = \frac{f_{m,output}}{\beta} + \frac{f_{m,output}}{\gamma} \tag{6}$$

where $f_{m,output}$ is the data size of the acquiring results for computing r_m, β is the data transmission rate from edge servers to base stations, γ is the data transmission rate from base stations to cloud data center.

The total time consumption for calculate all computing task is measured by

$$TC = \sum_{m=1}^{M} (g_m + h_m + b_m + o_m) \tag{7}$$

3.2 Load balance analysis

The resource utilization is a crucial element to evaluate the computing ability of the edge servers. The aim of this paper is improving the average resource utilization.

To ensure successful transmission for multiple functions of monitoring terminals, the resource allocation of VM on data links should be balanced. Therefore, the resource utilization of sw can be represented by

$$U_w = \frac{1}{a_w} \sum_{j=1}^{J} \sum_{l=1}^{a_w} qv_{j,l} \tag{8}$$

where $qv_{j,l}$ judges whether t_j occupied v_l in the s_w.

$$qv_{j,l} = \begin{cases} 1, \text{ if } t_j \text{ occupied } v_{w,l}, \\ 0, \text{ otherwise.} \end{cases} \tag{9}$$

Based on the number of servers which are occupied, we can calculate the average resource usage of edge servers. The number of occupied servers is defined by

$$OS = \sum_{w=1}^{W} qs_w \tag{10}$$

The average resource utilization of all edge servers is calculated by

$$U = \frac{1}{OS} \cdot \sum_{w=1}^{W} U_w \tag{11}$$

where qs_w judges whether s_w is occupied

$$qs_w = \begin{cases} 1, \text{ if } s_w \text{ is occupied,} \\ 0, \text{ otherwise.} \end{cases} \tag{12}$$

The load balance variance of s_w is calculated by

$$B_w = \frac{1}{(U_w - U)^2} \tag{13}$$

The average load balance variance of all edge servers is calculated by

$$B = \frac{1}{OS} \cdot \sum_{w=1}^{W} B_w \cdot qs_w \tag{14}$$

3.3 Problem formulation

In this paper, we aim to achieve the goal of minimizing the time consumption presented in (7) and obtaining load balance in (14). The formalized problem is given as

$$\min (TC), \tag{15}$$

$$\min (B), \tag{16}$$

$$s.t. \quad 0 \leq \sum_{w=1}^{W} qs_w \leq M \tag{17}$$

4 An Edge Computing-Enabled Resource Provisioning method for Video Surveillance in Internet of Vehicles

In summary, the goal of this paper is willing to solve this multi-objective optimization problem with minimizing the time consumption and optimizing load balance of video surveillance system in Internet of Vehicles at the same time. Compared with common genetic algorithms (GA) and evolutionary algorithms (EA), SPEA2, the method we choose eventually, with its superior performance and better robustness, is widely used in dealing with this kinds of problems to obtain a number of optimal strategies. Then the following job is to select the optimal solutions by using Simple Additive Weighting (SAW) and Multiple Criteria Decision Marking (MCDM) methods.

4.1 Encoding

In this chapter, the computing tasks in ENs are encoded firstly. In the GA, the value of the decision variable, i.e., the strategies for resource provisioning, are represented through gene. Ultimately a set of genes converge to a chromosome which represents optimal solution of the multi-object problem.

4.2 Fitness Functions and Constraints

The fitness functions in GA, which are used as decision criterion to evaluate the pros and cons of each individual. In this paper, the fitness functions include two different categories: the time consumption and the load balance of edge serverswhich represented respectively by (8) and (14). As is shown in (15), the design purpose of the method is to optimize the average resource utilization of edge servers and reduce the time consumption of computing tasks at the same time. Furthermore, the constraints are demonstrating in (16).

4.3 Initialization

In the initialization operation, the related parameters should be determined at first, which including the size of population SP, the probability of crossover PC, the probability of mutation PM, the number of iterations T and the size of archive SA. Each chromosome in GA represents the resource provisioning strategies in the population set which denoted $US^i = (US_1, US_2, US_3, \cdots, US_j)$ and US^i represents the i-th chromosome.

4.4 Selection

In the selection operation of SPEA2 algorithm, individuals who has more desirable fitness are selected from the current evolutionary group and placed into the mating pool. Hence the crossover and the mutation operation will select individuals from the mating pool only to generate a better population.

4.5 Crossover and Mutation

In the crossover operation of SPEA2 algorithm, two different parental chromosomes are combined to generate new chromosomes with better performance. In the first place, a crossover point in the parental chromosomes is picked in the crossover operation, then on both sides of this point, two genes are interchanged. Eventually, two new chromosomes are created around this point.

Mutation operation of SPEA2 algorithm takes place when the premature convergence emerges due to the descendant chromosome performs no longer more outstanding than their last generation but still not approach the satisfactory optimal solution. This vital operation is utilized to ensure the diversity of different individual and the equal mutation probability of each gene.

4.6 Optimal Strategy Selection by Using SAW and MCDM

The resource provisioning method this paper is dedicated to achieving a dynamic tradeoff between the two intent objective function E.g. time consumption and load balance in this paper. Meanwhile, both the time consumption and the load balance which given by the target problem of this paper are negative criteria. In other words, if the execution time of computing tasks getting longer, the result turns into more undesirable. Consequently, when the objective problem demands normalizing resource-scheduling indicator, which is the performance of time consumption and average resource utilization, that where SAW and MCDM take place. The load balance value is denoted as $LB = (LB^i, 1 \leq i \leq I)$, and the time consumption value is denoted as TC = $(TC^i, 1 \leq i \leq I)$. The chosen result is calculated by

$$OLB_{i,j} = \begin{cases} \frac{LB^{\max} - LB^{i,j}}{LB^{\max} - LB^{\min}} & , LB^{\max} - LB^{\min} \neq 0, \\ 1 & , LB^{\max} - LB^{\min} = 0. \end{cases} \tag{18}$$

$$OTC_{i,j} = \begin{cases} \frac{TC^{\max} - TC^{i,j}}{TC^{\max} - TC^{\min}} & , TC^{\max} - TC^{\min} \neq 0, \\ 1 & , TC^{\max} - TC^{\min} = 0. \end{cases} \tag{19}$$

where LB^{max},LB^{min},TC^{max} and TC^{min} represent the maximum value of load balance, the minimum value of load balance, the maximum value of time consumption and the minimum value of time consumption, respectively. Eventually, we calculate the utility value by

$$OR_i = OLB_{\mathrm{i,j}} \cdot tl + OTC_{i,j} \cdot tw, \tag{20}$$

where tl, tw represent the weight of the load balance and the time consumption, respectively.

4.7 Method Review

The final destination of objective question of this paper is to optimize the average load balance and the time consumption of all computing tasks deployed in edge servers. Due to the resource provisioning problem is a multi-objective optimization problem essentially, with its excellent performance we select SPEA2 algorithm to solve our target problem finally by means of comparing it with other GA. Above all, the objective problem is encoded. Then we proposed all of the fitness functions and the constraints to prepare for use. Furthermore, via implementing the environmental selection operation, chromosomes with more desirable performance of fitness are picked from their evolutionary group into the mating pool. Shortly afterwards, the crossover operation and the mutation operation is carried out to against the premature convergence and generate new individuals with more outstanding descendants. eventually, normalization processing of resource utilization and time consumption is carried out by ways of SAW and MCDM methods.

Algorithm 1 computing the optimal strategy by using SPEA2

Require: T
Ensure: OR

```
for i = 1 to I do
    t=1
    while t ≤ T do
        Calculate the total time consumption TC by (1) – (7)
        Calculate the average load balance variance LB by (8) – (14)
        Environmental selection to ensure the amount of OR
        Crossover and mutation operations to ensure the offspring
        t=t+1
    end while
    Evaluate utility value with SAW and MCDM method by(18)(19)
    select the optimal solution by (20)
end for
return OR
```

In summary, we proposed the procedure of obtaining the optimal strategy, the T represents the number of iterations and the OR represents the best strategy in the Algorithm 1.

5 EXPERIMENTAL EVALUATION

5.1 Comparison of employed ENs

With more ENs is employed, the time consumption and the load balance are affected soon. As an important parameter, the number of employed ENs must be taken into consideration. What's more, with the increase of the computing tasks, the more ENs will be employed. Fig.1 illustrates the number of ENs employed by the different resource provisioning methods. Apparently, the method ERPM occupied less ENs in dealing with same amount of computing tasks.

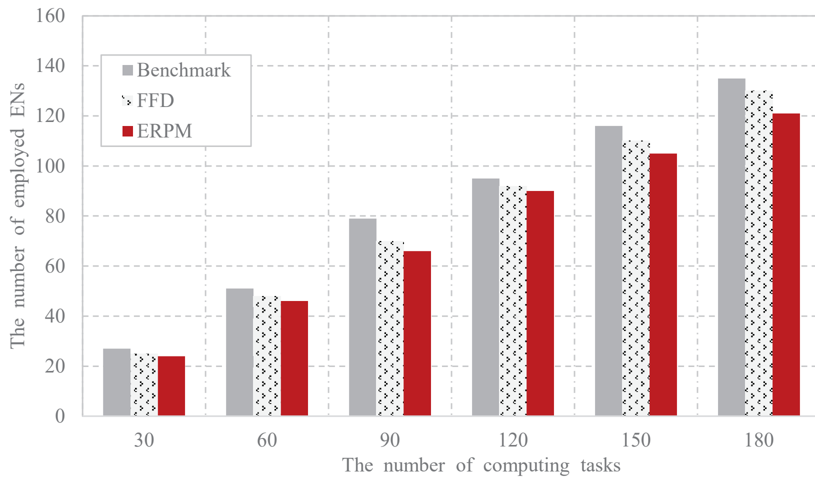

Fig. 1: Comparison of the number of employed ENs by different task scales.

5.2 Comparison of time consumption

After all computing tasks of processed and transmitted, the total time consumption is definitely achieved. The number of the time consumption has great importance in the whole computing tasks. Fig.2 shows the comparison of total time consumption of the ENs by using Benchmark, FFD and ERPM at different scales of computing tasks. The time consumption is calculated according to the executing process, the tasks migration and the offloading data. The value of time consumption is compared in Fig.2. It shows that the method ERPM has better performance at time consumption, that is to say, our proposed method ERPM based on SPEA2 wastes fewer time than the other methods.

5.3 Comparison of load balance

The load balance is a criterion to evaluate the resource utilization of ENs. It aims to optimize resource use, minimize response time, maximize throughput, and avoid overload of any single resource. Load balance improves the distribution of workloads across multiple computing resources In the method we proposed in this paper, six datasets of load balance are used to assess the pros and cons of our method. Fig.3 illustrates the comparison of the load balance of the ENs by

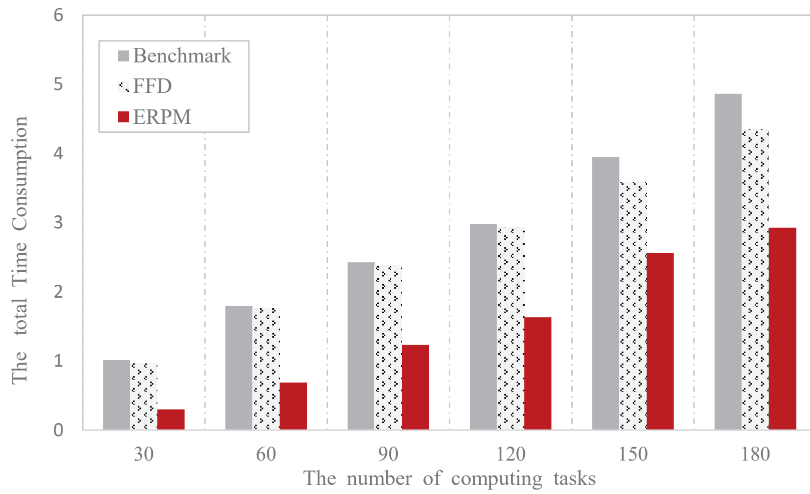

Fig. 2: Comparison of the time consumption by different task scales.

using Benchmark, FFD and our ERPM at different scales of computing tasks. Fewer load balance of employed ENs with more employed resource units yield a higher resource utilization and better performance. It is intuitive from Fig.3 that the method ERPM we proposed achieves better load balance than the other two offloading methods, which means to some extent, our method ERPM could reduce the overload or underload of ENs.

6 Conclusion

As a significant technology of information age, great influence is generated by video surveillance in Internet of Vehicles, not to mention other aspects of everyone's life in our land. Hence it is extremely urgent and important to optimize the time consumption and the load balance of all the ENs for edge computing-enabled resource provisioning for Video Surveillance in Internet of Vehicles. A better tasks computation offloading method is proposed and practiced in our paper. With its more desirable performance and the nature of offloading problem the SPEA2 algorithm is picked to achieve our goal. At the beginning, we analyze the details of process of tasks computing, then we formulate the problem as a multi-objective optimization problem. finally, we select an Evolution algorithm with simulation experimental to solve the objective problem we proposed.

In the future work, we are going to complicate our problem to adopt to the real world better and keep improve our algorithm with experiment.

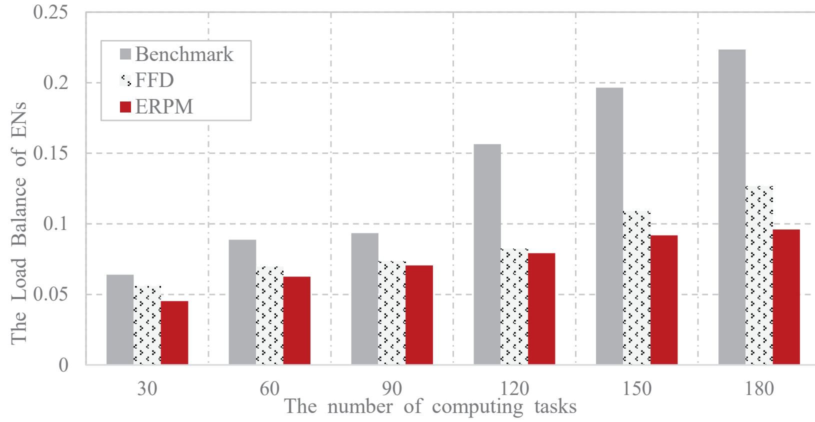

Fig. 3: Comparison of the load balance of the ENs by different scales.

Acknowledgment

This research is supported by the National Science Foundation of China under grant no. 61702277 and 61872219.

References

1. L. Atzori, A. Iera, and G. Morabito, "The internet of Things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
2. Zhanyang Xu, Renhao Gu, Tao Huang, Haolong Xiang, Xuyun Zhang, Lianyong Qi, and Xiaolong Xu. "An IoT-Oriented Offloading Method with Privacy Preservation for Cloudlet-Enabled Wireless Metropolitan Area Networks," Sensors 18, no. 9 (2018): 3030.
3. N. Kumar, J.J. Rodrigues, N. Chilamkurti, "Bayesian coalition game as-a-service for content distribution in internet of vehicles, " IEEE Internet Things J. 1 (6) (2014) 544–555.
4. Udaya L. N. Puvvadi, Kevin Di Benedetto, Aditya Patil, Kyoung-Don Kang, "Cost-Effective Security Support in Real-Time Video Surveillance," IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 11, NO. 6, DECEMBER 2015.
5. Changchun Long, Yang Cao, "Edge Computing Framework for Cooperative Video Processing in Multimedia IoT Systems," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 20, NO. 5, MAY 2018.
6. P. Lopez et al., "Edge-centric computing: Vision and challenges," ACMSIGCOMM Comput.Commun. Rev., vol. 45, no. 5, pp. 37–42, Sep. 2015.
7. Emil Eriksson, G. Dn, "Predictive Distributed Visual Analysis for Video in Wireless Sensor Networks," IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 15, NO. 7, JULY 2016.
8. Jie Zhang, Zhili Zhou, Shu Li, Leilei Gan, Xuyun Zhang, Lianyong Qi, Xiaolong Xu, and Wanchun Dou. "Hybrid computation offloading for smart home automation in mobile cloud computing," Personal and Ubiquitous Computing 22, no. 1 (2018): 121-134.

9. Jie Zhang, Lianyong Qi, Yuan Yuan, Xiaolong Xu, and Wanchun Dou. "A Workflow Scheduling Method for Cloudlet Management in Mobile Cloud,"2018 IEEE SmartWorld,DOI:10.1109/SmartWorld.2018.00167
10. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.
11. Lianyong Qi, Yi Chen, Yuan Yuan, Shucun Fu, Xuyun Zhang, Xiaolong Xu, A QoS-Aware Virtual Machine Scheduling Method for Energy Conservation in Cloud-based Cyber-Physical Systems, World Wide Web journal , 2019, DOI: 10.1007/s11280-019-00684-y.
12. Lianyong Qi, Qiang He, Feifei Chen, Wanchun Dou, Shaohua Wan, Xuyun Zhang, Xiaolong Xu,"Finding All You Need: Web APIs Recommendation in Web of Things through Keywords Search. IEEE Transactions on Computational Social Systems," 2019, DOI: 10.1109/TCSS.2019.2906925.
13. Po-Han Wu, Chih-Wei Huang, Jenq-Neng Hwang, "Video-Quality-Driven Resource Allocation for Real-Time Surveillance Video Uplinking Over OFDMA-Based Wireless Networks," IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 64, NO. 7, JULY 2015.
14. Jianguo Chen, Kenli Li, "Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing," IEEE Transactions on Industrial Informatics, DOI 10.1109/TII.2019.2909473.
15. Udaya L. N. Puvvadi, Kevin Di Benedetto, Aditya Patil, Kyoung-Don Kang, "Cost-Effective Security Support in Real-Time Video Surveillance," IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 11, NO. 6, DECEMBER 2015.
16. Xiaolong Xu, Yuancheng Li, Tao Huang, Yuan Xue , Kai Penge, Lianyong Qi f, Wanchun Dou, "An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks," Journal of Network and Computer Applications 133 (2019) 75–85.
17. Al-Nadwi, Musaddiq Majid Khan, et al. "Cloud Enabled e-Glossary System: A Smart Campus Perspective." International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage. Springer, Cham, 2018.
18. Yang, Jingyue, et al. "BDCP: A Framework for Big Data Copyright Protection Based on Digital Watermarking." International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage. Springer, Cham, 2018.