Trust-Aware Service Offloading for Video Surveillance in Edge Computing Enabled Internet of Vehicles

Xiaolong Xu, *Member, IEEE*, Qi Wu, Lianyong Qi, *Member, IEEE*, Wanchun Dou, *Member, IEEE*, Sang-Bing Tsai, and Md Zakirul Alam Bhuiyan, *Senior Member, IEEE*

***Abstract*— Internet of Vehicles (IoV) supports multiple traffic services by processing abundant data from sensors and video surveillance devices. With edge computing, video surveillance services can be certainly improved due to the handy resource provision for video storage and processing. Generally, to reduce the hardware and maintenance investment, it is a popular manner to deploy the limited amount of edge nodes along with the surveillance devices. However such edge node layout leads to the unstable service distribution and complicated data transmission across the surveillance devices and edge nodes, which consequently decreases the quality of the surveillance services. In addition, the service trustworthiness is suspected since the privacy information may be revealed to some extent during the data transmission. To combat these challenges, a trust-aware task offloading method (TOM) for video surveillance in edge computing enabled IoV is presented for minimizing the response time of the services, achieving the load balance of the edge nodes and realizing privacy protection. Technically, SPEA2 (improving the strength Pareto evolutionary algorithm) is employed to acquire balanced task offloading solutions. Then, TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) and MCDM (Multiple Criteria Decision Making) are exercised to ascertain the optimal solution. Finally, the experimental simulation demonstrates that TOM performs efficient and trust.**

***Index Terms*— IoV, edge computing, privacy protection, task offloading, SPEA2.**

Manuscript received December 17, 2019; revised February 12, 2020 and March 18, 2020; accepted April 9, 2020. Date of publication June 25, 2020; date of current version March 1, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1400600, in part by the Development Project of Jiangsu Province under Grant BE2019104, in part by the National Natural Science Foundation of China under Grant 61672276, Grant 61702277, and Grant 61872219, and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund. The Associate Editor for this article was H. Gao. *(Corresponding author: Wanchun Dou.)*

Xiaolong Xu is with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China, also with the Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China, also with the Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology and Engineering, Nanjing 210044, China, and also with the Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology and Engineering, Nanjing 210044, China (e-mail: njuxlxu@gmail.com).

Qi Wu is with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: wuqinuist@gmail.com).

Lianyong Qi is with the School of Information Science and Engineering, Qufu Normal University, Qufu 276826, China (e-mail: lianyongqi@gmail.com).

Wanchun Dou is with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: douwc@nju.edu.cn).

Sang-Bing Tsai is with the China Zhongshan Institute, University of Electronic Science and Technology, Zhongshan 528400, China, and also with the Research Center for Environment and Sustainable Development of the China Civil Aviation, Civil Aviation University of China, Tianjin 300300, China (e-mail: sangbing@hotmail.com).

Md Zakirul Alam Bhuiyan is with the Department of Computer and Information Sciences, Fordham University, New York, NY 10458 USA (e-mail: m.bhuiyan.dr@ieee.org).

Digital Object Identifier 10.1109/TITS.2020.2995622

I. Introduction

CURRENTLY, with the development of communication and networking technologies, the Internet of Things (IoT) is outstanding in the aspect of the connection between objects and humans. As the vehicles continue being connected to the IoT, the conventional vehicle Ad-hoc networks are changing into Internet of Vehicles (IoV). IoV enables the sensors and the video devices to receive the vehicle data which consists of position, vehicle-status and road-condition. Such collected information is used to process image, analyze traffic situation and recognize traffic accidents [1].

Among IoV related techniques, the video surveillance has a crucial impact on obtaining visual data in public for the IoV systems. Most current video surveillance systems employ the traditional cloud-based centralized solution for accommodating the massive all-day-operating video data, since the surveillance devices cannot host the abundant data. Although the cloud computing paradigm provides scalable resources for accommodating the video data, huge data communication overhead and latency limitation become key bottlenecks for real-time IoV services.

To acquire the prompt services, the distributed solution has been investigated on the video surveillance terminals [2]. As a popular distributed computing paradigm, edge computing has efficient storage resources, computing capacity and network connectivity around the video surveillance devices, which definitely enhances the responsive time of the video services [3], [4]. With edge computing, the video surveillance systems extend the communication and computation capabilities to deal with a wide range of computing tasks including video compressing, preprocessing and analyzing. Generally, benefiting from the short range wireless communication, the computing tasks are capable of being transferred across multiple devices to acquire the real-time task offloading, which promotes the overall performance of the video surveillance system [5], [6].

However, as imperative computing tasks gain, the limited resources in some edge nodes cannot dispose the glut of computing tasks, which will cause some tasks to be queued for execution at the edge nodes [7], [8]. Meanwhile, abundant computing resources may keep idle or non-busy status. In this case, it is necessary to guarantee the load balance of the edge nodes for the real-time response of the video services in the video surveillance systems [**?**], [9]. In addition, the edge nodes are easy attacked for illegally embezzling and tampering the video data, because the video data often have substantial redundancy in the edge nodes [11], [12]. Thus, the security and privacy during data offloading for the surveillance devices should be handled properly in the edge computing [13]. With these observations, we present a task offloading method (TOM) for trust-aware video surveillance in edge computing enabled IoV.

The key contributions of this paper are summarized as follows:

- The privacy entropy model is presented to handle the problem of privacy protection by a quantitative method of privacy.
- Adopt an algorithm named SPEA2 (improving the Strength Pareto Evolutionary Algorithm) [14] with simulation experimental to deal with the multi-objective problem.
- To confirm the final task offloading strategy, the technique for order preference by similarity to ideal solution (TOPSIS) and the multiple criteria decision making (MCDM) technique are leveraged, focusing on normalization and modeling of the aggregating function.
- A large amount of experimental consequences prove the effectiveness of TOM by performing multiple experiments with a Nanjing traffic monitor instance.

The remainder of this paper is organized as follows. In II Section, related work is discussed. Furthermore, system model and problem formulation in Sections III. A task offloading method for trust-aware video surveillance in edge computing enabled IoV is provisioned in Section IV. After that, experimental evaluation is presented in Section V. Finally, conclusion and future work are drawn in Section VI.

II. Related Work

In IoV, the edge computing affords the video surveillance devices the abilities of computing and storage. While the capacity of the surveillance systems is strengthen from the edge computing, some challenges incurred by the edge computing still exist. The time cost, the task offloading and the privacy preserving in edge computing are investigated respectively in the recent literatures.

The task offloading aims to appropriately partition the computing tasks and effectively portion them to the adjacent edge nodes [15]–[17]. In view of task offloading, an effective real-time video framework is formulated in [18] with efficient resource-provisioning strategy and low-delay video cluster scheduler to enhance utility, transmission efficiency, and image quality. In [19], an architecture for joint video processing was investigated to achieve the balance between finite network bandwidth and task offloading. Both of frameworks handle the problem of rare resource well. Meanwhile, it is crucial to consider the efficient of task offloading. In [20], Tziritas et.al. proposed an algorithm using hyper-graph segmentation to minimize the network overhead of systems with an optimal way in consideration of the unconstrained case. In consideration of offloading capacity, in [21], Chen et.al. developed a new network offloading framework to enhance the capacity of the edge computing, aiming at maximizing the persistent performance while guaranteeing the power cost with persistent constraints. In [22], Dinh et.al. proposed a offloading model to strengthen long-term utilities basing on a powerful distributed offloading model during various wirelesszhang2018data [11], [23].

In addition, in view of time cost, Chen et.al. presented a distributed smart surveillance system which transfers computing workloads to release the tremendous pressure of communication overheads and furnish the solutions of low-latency video analysis by deep learning algorithms in [24]. Tan and Hu *et.al.* [25] proposed a collaborative video processing strategy to achieve time-sensitive multimedia IoT assignments and enhance the human detection accuracy in edge computing.

Nevertheless, none of the above solutions consider the aspect of security features. The edge nodes in video surveillance face severe privacy preserving challenges, as it is difficult to guard against attack and leakage. To deal with this challenge, in [26], Kang et.al. presented a fog computing supported IoV (F-IoV) paradigm for resources exploiting to provide secure communication and privacy preservation, which effectively improves the location privacy of vehicles and releases the communication overhead in IoV systems. In [27], Puvvadi et.al. presented a novel protocol that weakens the latency for executing tasks, increases the supported bit rate, and provides desirable security features. A block size fitness scheme in the video surveillance was presented to optimize the problems of task offloading [28], which improves collaboration among servers, video devices and users. In addition, in [29], Li et.al. presented a data clustering strategy with privacy protecting for IoT applications, which guarantees data privacy of the terminals and provides source authentication in mobile edge computing. In [30], Zhou et.al. presented a privacy algorithm to preserve the data privacy, and designed a trust evaluation mechanism to guarantee accurately trustworthy in edge computing. With the aim of balancing among privacy-protecting and trustworthiness, the framework considers the edge computing structure to enhance the execution [31], [32].

However, current researches about the task offloading in edge computing mainly center around the implementation efficiency of the computing tasks, neglecting the shortcomings of the unbalanced service layout. In this paper, we try to develop a novel task offloading method to optimize both the running performance of the edge system and the responsive time of the computing tasks.

III. System Model

In this section, the time cost model, the load balance model and the privacy entropy model in IoV are designed and expounded. The main notations are presented in Table I.

TABLE I
NOTATIONS AND DEFINITIONS

Notation	Definition
M	The amount of video surveillance devices
W	The number of edge nodes
R	The video surveillance devices set, $R = \{r_1, r_2, \cdots, r_M\}$
S	The edge nodes set, $S = \{s_1, s_2, \cdots, s_W\}$
L	The amount of virtual machines in s_w
J	The number of computing tasks
V_w	The virtual machines set in s_w, $V_w = \{v_{w,1}, v_{w,2}, \cdots, v_{w,L}\}$
T	The computing task set, $T = \{t_1, t_2, \cdots, t_J\}$
T_{total}	The total time cost of services
B	The average load balance variance
I	The privacy entropy

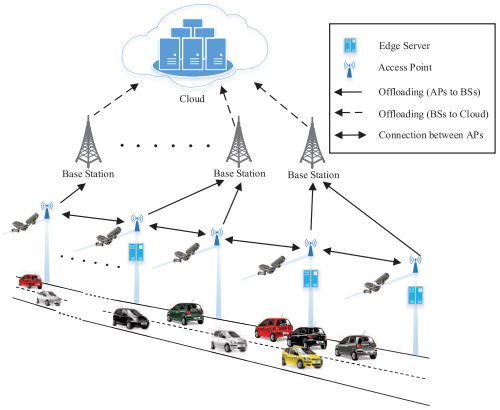


Fig. 1. A video surveillance architecture with the edge computing.

A. A Video Surveillance Architecture With Edge Computing

We employ edge computing technology in video surveillance in this paper. As indicated in Fig.1, an architecture of video surveillance with the edge computing is presented, where the surveillance terminals are installed on the poles of roadside and the generated video data are offloaded to the nearby edge nodes through access points (APs). The edge nodes with powerful computing ability copy with the complicated computing tasks. Then, the computing results are uploaded from the edge nodes to the cloud data center for the video analysis via the base stations. In this scenario, M poles are deployed along the roadside and a set of video surveillance device is fixed on each pole which is denoted as $R = \{r_1, r_2, \cdots, r_M\}$. A set of video surveillance device is constitutive of three components, i.e. surveillance terminals, edge node and APs. Owing to the high cost of station edge nodes, it is unreasonable for all the video surveillance devices to install the edge nodes. Hence, assume that $S = \{s_1, s_2, \cdots, s_W\}$ $(W < M)$ denote W edge nodes in video surveillance devices. Let $T = \{t_1, t_2, \cdots, t_J\}$ denote J computing tasks generating from the video terminals. The edge nodes possess powerful computing ability and storage capacity for preprocessing which consists of virtual machines (VMs). There are L VMs in s_w, defined as $V_w = \{v_{w,1}, v_{w,2}, \cdots, v_{w,L}\}$.

B. Time Cost Model

The time cost is comprised of the migration time between adjacent APs, the computing time at corresponding edge node and the offloading time. As a crucial parameter, the time cost decides the quality of real-time services in the surveillance devices.

When the collected video data from the surveillance device without edge node are accessed the nearby edge nodes, this process will generate the migration time which is calculated by

$$g_j = \sum_{m=1}^{M} \sum_{w=1}^{W} k_{j,m} k_{j,w} \frac{f_j}{\lambda} \sigma_{m,w}, \tag{1}$$

where λ is the data migration rate among APs, $\sigma_{m,w}$ is the times of the data transmission from r_m to r_w, k_j, m judges the $j-$th computing task t_j needs to be transmitted in r_m which is defined by

$$k_{j,m} = \begin{cases} 1, & t_j \text{ needs to be transmitted in } r_m, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

and k_j, w judges wether the t_j is transmitted to s_w which is defined by

$$k_{j,w} = \begin{cases} 1, & t_j \text{ is transmitted to } s_w, \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

When the video tasks are transfered to the objective node, the edge node should provide VMs to execute computation. The computing ability of edge nodes is related to the number of the idle VMs. Thus, the number of idle VMs in s_w is denoted as χ_w and the operational capacity of each VM is represented as ϕ. Assume that the video task with the length F_w is migrated to s_w, and the computation time of s_w is expressed as

$$h_j = \sum_{w=1}^{W} k_{j,w} \frac{F_j}{\chi_w \cdot \phi}. \tag{4}$$

The computing results (e.g. extracting feature) with small occupied capacity are offloaded from the edge nodes to cloud center. When the computing result of t_j with the weight $f_{j,output}$, the offloading time of t_j is calculated as

$$o_j = \sum_{w=1}^{W} k_{j,w} \left(\frac{f_{j,output}}{\eta} + \frac{f_{j,output}}{\lambda} \right), \tag{5}$$

where η denotes the transfer rate between the edge nodes and the base stations and λ denotes the transfer rate between the base stations and the cloud data center.

With the three components of time, the total time cost is measured by

$$T_{total} = \sum_{j=1}^{J} (g_j + h_j + o_j). \tag{6}$$

C. Load Balance Model

The load balance is an important element for assessing the utilization of resource in edge nodes. Load balance distributes the massive workloads to optimize resource utilization [34]. When the variance of load balance is smaller, the load balance of each computing node is more balanced.

Let $v_{w,l}$ represent the $l-$th ($l = \{1, 2, \cdots, L\}$) VM in the s_w ($w = \{1, 2, \cdots, W\}$). Firstly, the resource utilization of s_w is calculated as

$$U_w = \frac{1}{\chi_w} \sum_{j=1}^{J} \sum_{l=1}^{\chi_w} k_{j,w} \xi_j^{w,l}, \tag{7}$$

where $\xi_j^{w,l}$ judges wether t_j occupies $v_{w,l}$, denoted as

$$\xi_j^{w,l} = \begin{cases} 1, & \text{if } t_j \text{ occupies } v_{w,l}, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

With the amount of the occupied edge nodes, the average resource utilization of edge nodes is computed. Hence, the average resource utilization of all edge nodes is calculated by

$$U = \frac{1}{N_{occupied}} \sum_{w=1}^{W} U_w, \tag{9}$$

where $N_{occupied}$ represents the amount of the occupied edge nodes, calculated as

$$N_{occupied} = \sum_{w=1}^{W} \sum_{j=1}^{J} k_{j,w}. \tag{10}$$

Then, the average load variance is calculated as the reciprocal of the resource utilization variance, which is more reflective of the resource balance. The load variance of s_w is expressed as

$$B_w = \frac{1}{(U_w - U)^2}. \tag{11}$$

Finally, with the load variance of each node, the average load variance is expressed as

$$B = \frac{1}{N_{occupied}} \sum_{w=1}^{W} B_w k_{j,w}. \tag{12}$$

D. Privacy Entropy Model

In the privacy protection, the quantification of privacy plays a key role, which improves the ability of the risk assessment of the privacy disclosure. Furthermore, the information entropy as a quantitative model of information is leveraged to deal with the problem of the privacy measurement. The basic information entropy model of the privacy protection, called the privacy entropy model, is proposed for the quantification of privacy. In the model, the edge node is formulated as the privacy source, the privacy attacker is regarded as the recipient, and the privacy disclosure course is regarded as a communication channel. The mathematical model of the privacy source X is expressed as

$$\begin{pmatrix} X \\ Q(X) \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_s & \cdots & x_m \\ q(x_1) & q(x_2) & \cdots & q(x_s) & \cdots & q(x_m) \end{pmatrix}, \tag{13}$$

where $0 \le q(x_s) \le 1$ and $\sum_{s=1}^{m} q(x_s) = 1$,

Similarly, the mathematical model of the privacy attacker Y is expressed as

$$\begin{pmatrix} Y \\ Q(Y) \end{pmatrix} = \begin{pmatrix} y_1 & y_2 & \cdots & y_z & \cdots & y_k \\ q(y_1) & q(y_2) & \cdots & q(y_z) & \cdots & q(y_k) \end{pmatrix}, \tag{14}$$

where $0 \le q(y_z) \le 1$ and $\sum_{z=1}^{k} q(y_z) = 1$.

To describe the average private information of the privacy source, the privacy source entropy is defined as $H(X)$. The larger the privacy source entropy is, the less likely the privacy leak is. Hence, the privacy entropy is leveraged to measure the degree of privacy protection.

$$H(X) = -\sum_{s=1}^{m} q(x_s) \log_2 q(x_s). \tag{15}$$

When the privacy attackers are obtaining some private information, the privacy condition entropy $H(X/Y)$ is introduced as the uncertainty degree of the privacy that is defined as

$$H(X/Y) = -\sum_{s=1}^{m} \sum_{z=1}^{k} q(x_s y_z) \log_2 q(x_s/y_z). \tag{16}$$

The conditional entropy indicates that the privacy source X still has an uncertainty after receiving Y. The degree of uncertainty is caused by the interference of enemy in the process of observing the privacy source.

The average privacy information $I(X; Y)$ describes the degree of privacy leakage, which is defined as

$$I(X; Y) = \sum_{s=1}^{m} \sum_{z=1}^{k} q(x_s y_z) \log_2 \frac{q(x_s/y_z)}{q(x_s)}. \tag{17}$$

IV. TOM Design

In this section, firstly, the task offloading problem for video surveillance is defined. Then, TOM solves multi-objective optimization problem by SPEA2. Then, TOM implements the normalization processing by TOPSIS and MCDM methods.

A. Problem Definition

In this paper, the purposes include the minimum of the time cost by (6), the optimization of the load balance by (12) and the maximum of the privacy entropy by (17). The task offloading problem is regarded as the multi-objective optimal problem. The problem formulation is presented as

$$\min(T_{total}), \min(B), \max(I), \tag{18}$$

$$s.t. \sum_{s=1}^{m} q(x_s) = 1, \sum_{z=1}^{k} q(y_z) = 1, \tag{19}$$

where constraints indicate that the number of the video surveillance devices is larger than the number of the edge nodes, the probability of all privacy sources is equal to 1, and the probability of all privacy attractors is equal to 1.

B. Trust-Aware Task Offloading Method Using SPEA2

SPEA2 has excellent capacity and better robustness which is beneficial to deal with various problems to get the optimal strategy [35], [36]. Due to the efficient capacity of SPEA2 in handling the multi-objective problem, SPEA2 achieves the better performance in the task offloading problem. Hence, SPEA2 is superior to other genetic algorithms (GAs) and evolutionary algorithms (EAs), so we chose it as the optimal algorithms of TOM.

1) Encoding: The computing tasks collection in edge nodes are encoded firstly. The strategies for task offloading are formulated as chromosomes in the population. According to the situation that not all the video surveillance devices deploy the edge nodes, suppose that the edge nodes for executing tasks regard as genes and each gene k_w in the chromosome is set at $\{1, 2, \ldots, W\}$. Through optimal operation, the optimal solution is calculated as an superior individual.

2) Fitness Functions and Constraints: To assess the advantage and disadvantage of individuals, the fitness functions are used as decision criterion. The fitness functions comprise the time cost, the load balance and the privacy entropy which are represented respectively by (6), (12) and (17). Furthermore, (19) shows two constraints. Since not all the video surveillance devices deploy the edge nodes, the amount of the video surveillance devices is larger than the amount of the edge nodes. To ensure the reliability of privacy protecting, the probabilities of privacy sources and privacy attractors are limited.

3) Initialization: The related parameters are defined in the initialization operation. First of all, the individual represents the optimal solution and the length of individual represents the amount of tasks. What's more, the initial population set P has the population size N, the initial archive P_0 has the population size N_0 and the maximal iteration is Q. The population set of the q-th generation is denoted as $P(q, N)$. The archive set of the q-th generation is denoted as $P_0(q, N_0)$. Then, the probability of mutation is formulated as P_M and the probability of crossover is formulated as P_C.

4) Selection: The fitness individuals from the current evolutionary cluster are selected and the superior individuals are picked into the mating pool in the selection operation. $P(q, N)$ and $P_0(q, N_0)$ form the new archive set with N_1 size which is denoted as $P_0(q+1, N_1)$, which is performed through the selection operation. In the non-dominated sorting operation, the dominated individuals of P and P_0 should be eliminated, and the remnant individuals form the new archive set. If N_1 is larger than N_0, the crowded-comparison operation will be leveraged to cut down the size of the archive set. Through the selection operation, the total amount of individuals in P and P_0 is under N_P. Select the desirable individuals from $P(q, N)$ and $P_0(q, N_0)$ to generate the next generation archive set $P_0(q+1, N_1)$, and copy the new archive set to the new population set $P(q+1, N_1)$. Then, to generate an excellent population, single out the superior individuals through the crossover and the mutation operations.

5) Crossover and Mutation: To create novel chromosome with excellent property, the crossover operation of SPEA2 combines two diverse parental chromosomes through complex interchangeable steps. Firstly, a crossover site of the parental generation is selected with the probability P_C, and two related genes around this site are exchanged. Ultimately, two novel chromosomes are generated around this site. In addition, the mutation operation of SPEA2 comes up with the probability P_M with the gradual convergence, because the offspring did no better than their parents which cannot bring the satisfactory optimal solution. The diversity of different individuals is guaranteed through this operation.

C. Optimal Strategy Selection Using TOPSIS and MCDM

The multiple solutions are generated by SPEA2, and the next work is to generate the optimal solutions using TOPSIS and MCDM [37], [38]. Assume that there are C alternative strategies and each strategy includes three attributes, the total time cost, the average load balance variance and the privacy entropy which are denoted as $T_c = \{T_1, T_2, \cdots, T_C\}$, $B_c = \{B_1, B_2, \cdots, B_C\}$, and $I_c = \{I_1, I_2, \cdots, I_C\}$ $(c = \{1, 2, \cdots, C\})$ respectively. The procedure of the TOPSIS method is applied in TOM. The normalized decision matrix is calculated firstly. The normalized values of the total time cost D_c^T, the average load balance variance D_c^B and the privacy entropy D_c^I are denoted respectively as

$$D_c^T = T_c \left(\sum_{c=1}^{C} T_c^2 \right)^{-\frac{1}{2}}, \tag{20}$$

$$D_c^B = B_c \left(\sum_{c=1}^{C} B_c^2 \right)^{-\frac{1}{2}}, \tag{21}$$

and

$$D_c^I = I_c \left(\sum_{c=1}^{C} I_c^2 \right)^{-\frac{1}{2}}. \tag{22}$$

The weights of the total time cost, the average load balance variance and the privacy entropy are denoted as W_T, W_B, and W_I $(W_T + W_B + W_I = 1)$. The weighted normalized values of the total time cost E_c^T, the average load balance variance E_c^B and the privacy entropy E_c^I are calculated as

$$E_c^T = W_T D_c^T, \tag{23}$$

$$E_c^B = W_B D_c^B, \tag{24}$$

and

$$E_c^I = W_I D_c^I. \tag{25}$$

In TOM, the ideal and negative-ideal solutions are determined. Hence, the total time cost is determined as the ideal solution, while the average load balance variance and the privacy entropy are determined as the negative-ideal solutions. What's more, the benefit criteria with maximum and the cost criteria with minimum for solution are determined which are calculated as

$$A^* = \{E_{\max}^{T*}, E_{\min}^{B*}, E_{\min}^{I*}\}, \tag{26}$$

and

$$A^- = \{E_{\min}^{T-}, E_{\max}^{B-}, E_{\max}^{I-}\}, \tag{27}$$

where $E_{\max}^{T*}$, $E_{\min}^{B*}$, and $E_{\min}^{I*}$ are respectively the ideal solution of the total time cost, the negative-ideal solution of the average load balance variance and the negative-ideal solution of the privacy entropy in the benefit criteria. $E_{\min}^{T-}$, $E_{\max}^{B-}$, and $E_{\max}^{I-}$ are respectively the ideal solution of the total time cost, the negative-ideal solutions of the average load balance variance and the privacy entropy in the cost criteria.

The each alternative separation for the ideal solution is calculated by the dimensional Euclidean distance, which is given as

$$G_c^* = \sqrt{(E_c^T - E_{\max}^{T*})^2 + (E_c^B - E_{\min}^{B*})^2 + (E_c^I - E_{\min}^{I*})^2}, \quad (28)$$

Similarly, the each alternative separation for the negative-ideal solution is calculated by

$$G_c^- = \sqrt{(E_c^T - E_{\min}^{T-})^2 + (E_c^B - E_{\max}^{B-})^2 + (E_c^I - E_{\max}^{I-})^2}. \quad (29)$$

Then, the relative closeness of the alternative solution is given as

$$F_c^* = \frac{G_c^-}{G_c^* + G_c^-}. \quad (30)$$

All the alternatives are arranged in the order of relative closeness. Finally, the optimal solution F is calculated as

$$F = max[F_c^*], \quad (31)$$

$$s.t.\ W_T, W_B, W_I \in [0, 1],$$

$$W_T + W_B + W_I = 1. \quad (32)$$

In summary, we conduct normalization and modeling aggregating function by TOPSIS and MCDM in the Algorithm 1. Firstly, the inputs are the strategy of the total time cost, the average load balance variance and the privacy entropy, and the weights of the total time cost, the average load balance variance and the privacy entropy. This algorithm obtains the optimal strategy in each schedule F. The normalized decision matrices are calculated (Line 2) and then calculate the weighted normalized value (Line 3). What's more, the ideal and negative-ideal solutions are determined (Line 4) and the benefit criteria with maximum and cost criteria with minimum are determined (Line 5). Repeat this process until this iteration ends. Finally, output the optimal strategy F.

Algorithm 1 Selecting the Optimal Strategy

Require: T_c, B_c, I_c, W_T, W_B, W_I
Ensure: F
1: **for** $c = 1$ to C **do**
2: Calculate D_c^T, D_c^B and D_c^I by (20) to (22)
3: Calculate E_c^T, E_c^B and E_c^I by (23) to (25)
4: Determine the ideal and negative-ideal solution
5: Determine the benefit criteria with maximum and cost criteria with minimum
6: Calculate the alternative separation for the ideal and negative-ideal solution respectively by (28) and (29)
7: Calculate the relative closeness of F by (31)
8: **end for**
9: **return** F

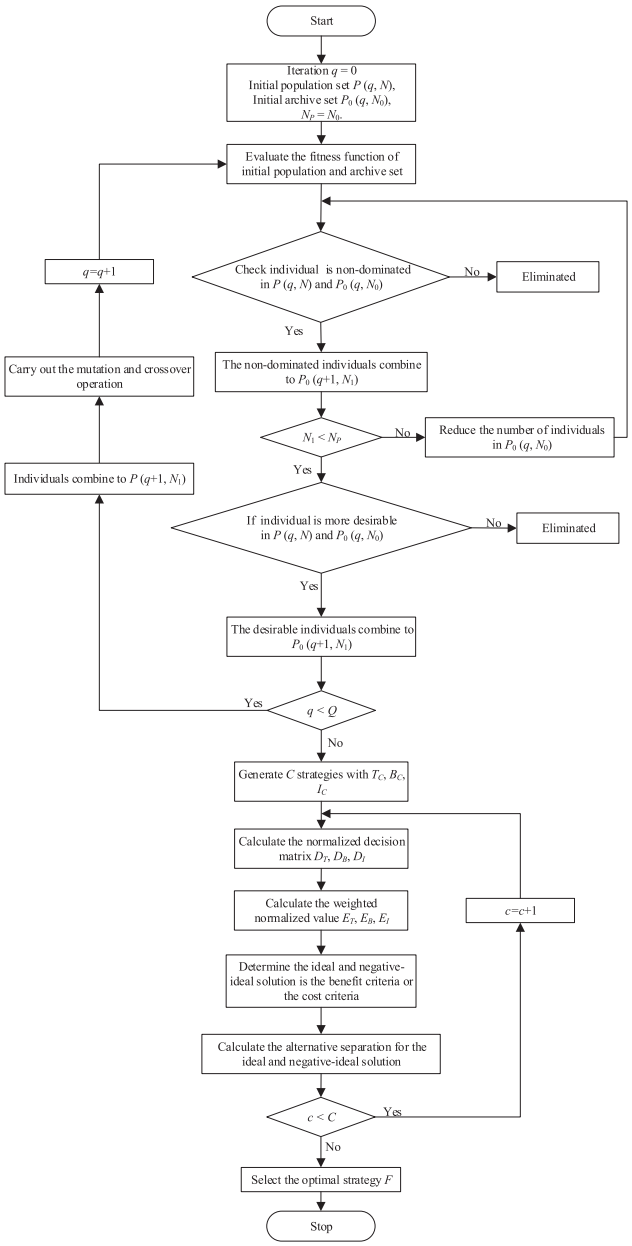


Fig. 2. The flow diagram of TOM.

D. Method Review

The final goal of task offloading is to shorten the time cost, the average load balance and the privacy entropy in edge nodes. First of all, the encoding operation abstracts genetic problem from the proposed model. What's more, the fitness functions are employed to assess individuals, and the constraints favor of the convergence of individuals. Then, during the selection operations, the superior individuals are selected in the mating pool. Furthermore, to generate novel individuals, the crossover and mutation operations ensure the diversity of individuals with more excellent offspring. Finally, normalization processing of time cost, load balance and privacy entropy is implemented by using TOPSIS and MCDM methods. The flow diagram of SPEA2 with TOPSIS and MCDM methods is shown in Fig.2.

TABLE II
PARAMETER SETTINGS

Parameter	Value
T	50, 100, 150, 200, 250
L	9
α	400Mbit/s
β	500Mbit/s
γ	600Mbit/s
p	[0.5GB, 0.8GB]

The overview of TOM of obtaining the optimal strategy is interpreted in Algorithm 2. First of all, the inputs of the algorithm are the size of population, the maximum iterations, the probability of mutation and crossover. This algorithm obtains the optimal strategy in each schedule F. The total time cost, the load balance and the privacy entropy are respectively calculated (Lines 4 to 8) and then select the optimal individuals to generate the next generation (Line 9). What's more, the utility values are evaluated with TOPSIS and MDCM (Line 13) and the optimal solutions are picked out (Line 14). Repeat this process until this iteration ends. Finally, output the optimal strategy F.

Algorithm 2 Resource Provisioning Method

Require: Q, N_P, P_C, P_M
Ensure: F
1: **for** $c = 1$ to C **do**
2: $q = 1$
3: **while** $q \leq Q$ **do**
4: **for** $n_0 = 1$ to N_P **do**
5: Calculate T_{total} by (6)
6: Calculate B by (12)
7: Calculate I by (17)
8: **end for**
9: Selection to guarantee the amount of the offspring generation
10: Crossover and mutation operations
11: $q = q + 1$
12: **end while**
13: Evaluate utility value using TOPSIS and MCDM method by **Algorithm 1**
14: Select the optimum solution by (31)
15: **end for**
16: **return** F

V. EXPERIMENTAL EVALUATION

A. Experimental Context

In the experiment, TOM is implemented in PC that has disposition with the Intel Core I5-8400H processor, 8 GB RAM and 250GB solid drive. The essential parameters and the scope of values are indicated in Table II. To guarantee the diversity and equity of the experiment, two methods are used to compare with TOM basing on the true dataset of video surveillance nodes in Nanjing as showing Fig.3.

-Benchmark: The waiting tasks are transferred to the adjacent edge nodes. If there is no enough idle space to execute next task in the former edge node, this task will be transferred to the latter edge node. The program is reiterated until that all the computing tasks are transferred.

-First Fit Decreasing (FFD): The task is deployed in the first fit edge node which is searched by traversal operation. If there is enough room to perform tasks in this edge node, the latter task will be migrated in it. This procedure is repeated until offloading ends.

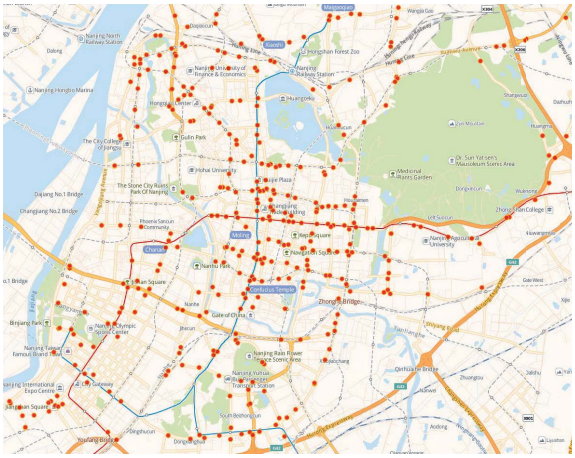


Fig. 3. The location distribution of video surveillance nodes in Nanjing on map.

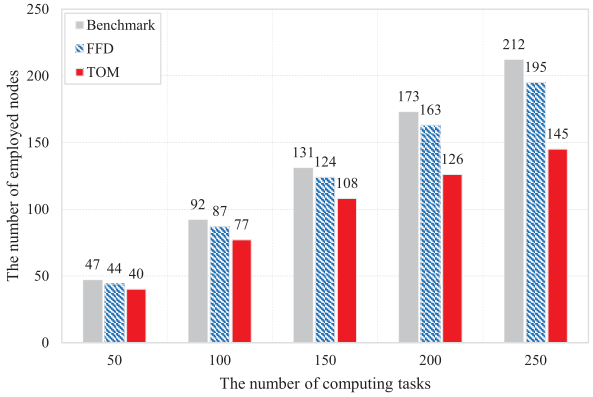


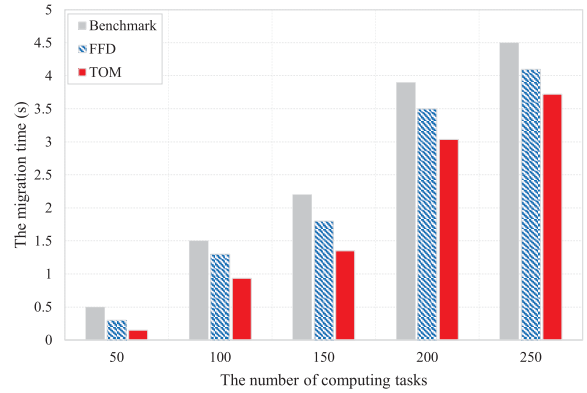
Fig. 4. Contrast on the number of employed nodes by Benchmark, FFD and TOM.

B. Contrast on Number of Employed Nodes

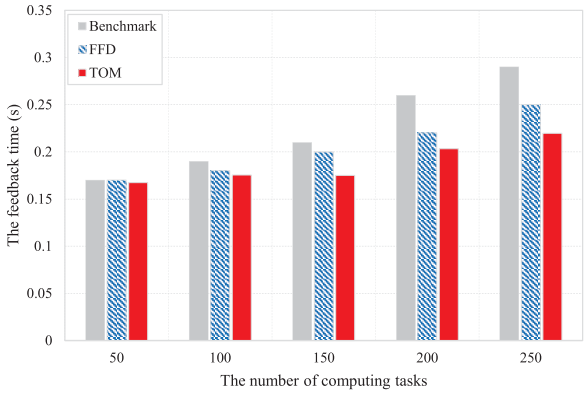
The number of employed edge nodes is used to authenticate the computing resource cost of the methods. Fig.4 illustrates the amount of nodes employed by the different task offloading solutions. For instance, when the amount of computing tasks is 250, the number of employed edge nodes with Benchmark, FFD and TOM are 212, 195 and 145 respectively. This histogram shows that TOM just occupies a handful of computing resource. The reason is that TOM is a multi-objective optimization method for achieving the global optimal solution, which makes fewer occupied nodes than the other two methods.

C. Contrast on Time Cost

The total time cost includes three components, namely the executing time, the transmission and the feedback. The migration time and feedback time should be considered firstly.



(b) The migration time



(a) The feedback time

Fig. 5. Contrast on the migration time and the feedback time by Benchmark, FFD and TOM.

The migration time and the feedback time are compared by using Benchmark, FFD and TOM in Fig.5.

After processing and transmitting, the total time cost is calculated. Fig.6 illustrates the contrast of total time cost with Benchmark, FFD and TOM with various scales of computing tasks. Fig.6 shows that the time cost of TOM is less than the other methods.

D. Contrast on Resource Utilization

Fig.7 illustrates the contrast of edge resource utilization of the nodes with different scales of computing tasks by using Benchmark, FFD and TOM. It is evident that TOM possesses higher resource utilization. With the same number of computing tasks, the number of occupied nodes with TOM is less than the other two methods, implying that the computing resource for processing tasks will be utilized efficiently. Hence, the resource utilization with TOM is better than both Benchmark and FFD.

E. Contrast on Load Balance

As a criterion, the load balance appraises the resource balance of the edge nodes. Fig. 8 depicts the load balance versus the number of computing tasks in three methods. TOM achieves a better performance than Benchmark and FFD. With the same number of tasks, the load balance with TOM is lower than the other two methods, which indicates that the situation of single resource overload across multiple computing resources is avoided to improve the balance distribution of

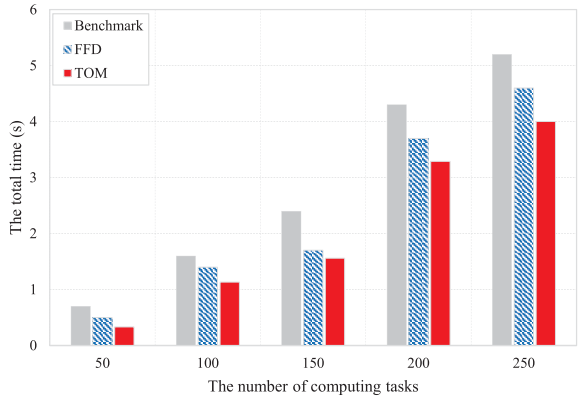


Fig. 6. Contrast on the total time cost by Benchmark, FFD and TOM.

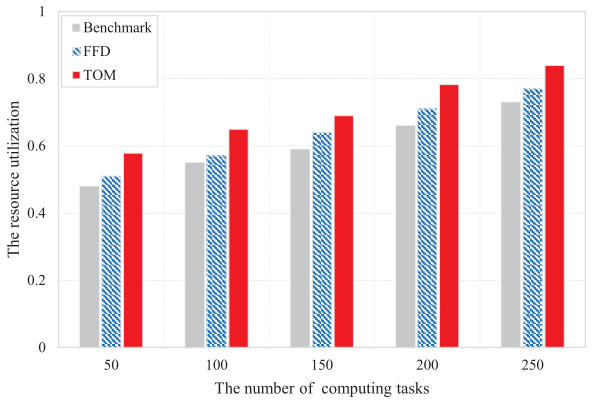


Fig. 7. Contrast on resource utilization by Benchmark, FFD and TOM.

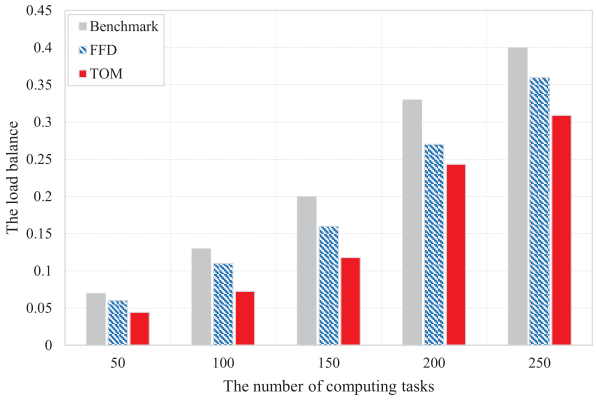


Fig. 8. Contrast on the load balance by Benchmark, FFD and TOM.

resource more effectively. For example, when the number of computing tasks is 200, the load balances of Benchmark and FFD are both above 0.25, while TOM is significantly lower than the value.

F. Contrast on Privacy Entropy

As an information entropy, the higher the value of privacy entropy is, the more secure the task transmission is. From Fig.9, the privacy entropy of Benchmark, FFD and TOM increases with the number of tasks. Meanwhile, when the number of tasks is small, the performance of TOM is almost the same as the other two methods. However, with the number of computing tasks increasing, the gap between the privacy entropy values of three methods is increasing. According to

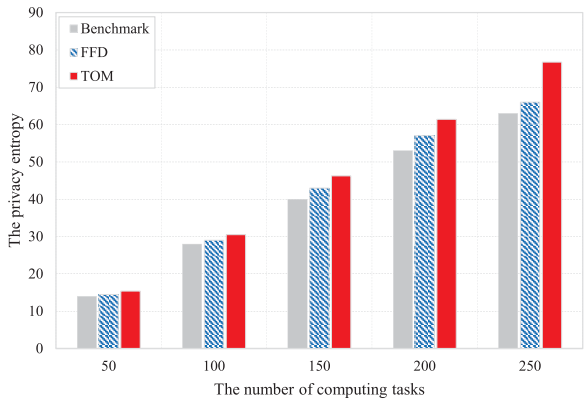


Fig. 9. Contrast on the privacy entropy by Benchmark, FFD and TOM.

experimental results, when the amount of computing tasks is 50, 100, 150, 200 and 250 respectively, the improving rates of FFD relative to Benchmark are 3.51%, 3.59%, 7.51%, 7.54% and 5.91%, and the improving rates of TOM relative to Benchmark are 9.17%, 8.91%, 15.03%, 15.86% and 21.72%. Therefore, when the number of tasks is huge, we deduce that TOM has stronger improvement on privacy pretection than FFD.

VI. Conclusion

A task offloading method for the video surveillance in edge computing enable IoV was proposed and practiced to shorten the time cost, keep the load balance of edge nodes and enhance the privacy protection. We analyzed the program of video tasks, then we formulated the problems of task offloading, time cost of services, and the privacy entropy as a multi-objective optimization problem. Owing to the excellent capacity, we chosen SPEA2 to solve the objective problem of tasks offloading. Then, TOPSIS and MCDM were used to ascertain the optimal solution. Eventually, simulation experimental with TOM is leveraged to figure out the multi-objective problem. In the future work, TOM is applied to the complex actual systems and improve TOM with experiment in real video surveillance systems.

References

[1] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of Internet of vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.

[2] X. Zhang *et al.*, "Enhancing video event recognition using automatically constructed semantic-visual knowledge base," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1562–1575, Sep. 2015.

[3] P. G. Lopez *et al.*, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, 2015.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[5] E. Eriksson, G. Dan, and V. Fodor, "Predictive distributed visual analysis for video in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1743–1756, Jul. 2016.

[6] B. Jo, M. Jalil Piran, D. Lee, and D. Young Suh, "Efficient computation offloading in mobile cloud computing for video streaming over 5G," *Comput., Mater. Continua*, vol. 61, no. 2, pp. 439–463, 2019.

[7] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018.

[8] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.

[9] X. Xue, S. Wang, L. Zhang, Z. Feng, and Y. Guo, "Social learning evolution (SLE): Computational experiment-based modeling framework of social manufacturing," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3343–3355, Jun. 2019.

[10] T. Wang, M. Z. A. Bhuiyan, G. Wang, L. Qi, J. Wu, and T. Hayajneh, "Preserving balance between privacy and data integrity in edge-assisted Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2679–2689, Apr. 2020.

[11] Y. Guo, F. Liu, N. Xiao, and Z. Chen, "Task-based resource allocation bid in edge computing micro datacenter," *Comput., Mater. Continua*, vol. 61, no. 2, pp. 777–792, 2019.

[12] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy efficient dynamic offloading in mobile edge computing for Internet of Things," *IEEE Trans. Cloud Comput.*, early access, Feb. 19, 2019, doi: 10.1109/TCC.2019.2898657.

[13] X. Xue, H. Han, S. Wang, and C.-Z. Qin, "Computational experiment-based evaluation on context-aware O2O service recommendation," *IEEE Trans. Services Comput.*, vol. 12, no. 6, pp. 910–924, Nov. 2019.

[14] G. Yu, T. Chai, and X. Luo, "Multiobjective production planning optimization using hybrid evolutionary algorithms for mineral processing," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 487–514, Aug. 2011.

[15] Q. He *et al.*, "A game-theoretical approach for user allocation in edge computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 515–529, Mar. 2020.

[16] P. Lai *et al.*, "Optimal edge user allocation in edge computing with variable sized vector bin packing," in *Proc. Int. Conf. Service-Oriented Comput.* Hangzhou, China: Springer, 2018, pp. 230–245.

[17] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4831–4843, Jun. 2019.

[18] P.-H. Wu, C.-W. Huang, J.-N. Hwang, J.-Y. Pyun, and J. Zhang, "Video-quality-driven resource allocation for real-time surveillance video uplinking over OFDMA-based wireless networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 7, pp. 3233–3246, Jul. 2015.

[19] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, early access, Mar. 4, 2019, doi: 10.1109/TETC.2019.2902661.

[20] N. Tziritas *et al.*, "Data replication and virtual machine migrations to mitigate network overhead in edge computing systems," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 4, pp. 320–332, Oct. 2017.

[21] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1619–1632, Aug. 2018.

[22] T. Quang Dinh, Q. Duy La, T. Q. S. Quek, and H. Shin, "Learning for computation offloading in mobile edge computing," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6353–6367, Dec. 2018.

[23] M. Zhang, C. Chen, T. Wo, T. Xie, M. Z. A. Bhuiyan, and X. Lin, "SafeDrive: Online driving anomaly detection from large-scale vehicle data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2087–2096, Aug. 2017.

[24] J. Chen, K. Li, Q. Deng, K. Li, and P. S. Yu, "Distributed deep learning model for intelligent video surveillance systems with edge computing," *IEEE Trans. Ind. Informat.*, early access, Apr. 9, 2019, doi: 10.1109/TII.2019.2909473.

[25] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10190–10203, May 2018.

[26] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported Internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2627–2637, Aug. 2018.

[27] U. L. N. Puvvadi, K. D. Benedetto, A. Patil, K.-D. Kang, and Y. Park, "Cost-effective security support in real-time video surveillance," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1457–1465, Dec. 2015.

[28] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Jan. 2019.

[29] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4755–4763, Jun. 2019.
[30] P. Zhou, K. Wang, J. Xu, and D. Wu, "Differentially-private and trustworthy online social multimedia big data retrieval in edge computing," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 539–554, Mar. 2019.
[31] Y. Ding, L. Tian, B. Han, H. Wang, Y. Wang, and J. Xi Zheng, "Achieving privacy-preserving iris identification via el gamal," *Comput., Mater. Continua*, vol. 61, no. 2, pp. 727–738, 2019.
[32] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: Task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Trans. Cloud Comput.*, early access, Jun. 20, 2019, doi: 10.1109/TCC.2019.2923692.
[33] T. Wang, J. Zhou, A. Liu, M. Z. A. Bhuiyan, G. Wang, and W. Jia, "Fog-based computing and storage offloading for data synchronization in IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4272–4282, Jun. 2019.
[34] J. L. J. Laredo, F. Guinand, D. Olivier, and P. Bouvry, "Load balancing at the edge of chaos: How self-organized criticality can lead to energy-efficient computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 2, pp. 517–529, Feb. 2017.
[35] S. Jiang, J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202–2213, Oct. 2015.
[36] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, Apr. 2014.
[37] G Büyüközkan and G. Çifçi, "A novel hybrid MCDM approach based on fuzzy DEMATEL, fuzzy ANP and fuzzy TOPSIS to evaluate green suppliers," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3000–3011, Feb. 2012.
[38] M. Behzadian, S. K. Otaghsara, M. Yazdani, and J. Ignatius, "A state-of the-art survey of topsis applications," *Expert Syst. Appl.*, vol. 39, no. 17, pp. 13051–13069, 2012.

Xiaolong Xu (Member, IEEE) received the Ph.D. degree from Nanjing University, China, in 2016. He worked as a Research Scholar with Michigan State University, USA, from April 2017 to May 2018. He is currently an Assistant Professor with the School of Computer and Software, Nanjing University of Information Science and Technology. He has published more than 80 peer-reviewed papers in the international journals and conferences, including the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON BIG DATA, the IEEE INTERNET OF THINGS, the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, the *Journal of Network and Computer Applications*, *Society of Petroleum Engineers*, WWWj, IEEE ICWS, and ICSOC. His research interests include fog computing, edge computing, the Internet of Things, cloud computing, and big data.

Qi Wu received the B.S. degree in computer science and technology engineering from the Nanjing University of Information Science and Technology in 2019, where he is currently pursuing the master's degree in computer science and technology. His research interests include mobile computing, big data, cloud computing, and machine learning.

Lianyong Qi (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2011. He is currently an Associate Professor with the School of Information Science and Engineering, Chinese Academy of Education Big Data, Qufu Normal University, China. He has already published more than 50 articles, including the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, TBD, FGCS, the *Journal of Computational Social Science*, CCPE, ICWS, and ICSOC. His research interests include services computing, big data, and the Internet of Things.

Wanchun Dou (Member, IEEE) received the Ph.D. degree in mechanical and electronic engineering from the Nanjing University of Science and Technology, China, in 2001. He is currently a Lecturer with the Nanjing University of Science and Technology. He is also a Full Professor with the State Key Laboratory for Novel Software Technology, Nanjing University. From April 2005 to June 2005 and from November 2008 to February 2009, he visited the Departments of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, respectively, as a Visiting Scholar. He has published more than 100 research papers in international journals and international conferences. His research interests include workflow, cloud computing, and service computing.

Sang-Bing Tsai received the Ph.D. degree in technology management and business management. He is currently a Professor with the Zhongshan Institute, University of Electronic Science and Technology of China, and the Civil Aviation University of China. He has over 150 published peer-reviewed journal articles. His recent research interests include operation management, computer science, big data, and applied mathematics. He is the Co-Editor-in-Chief of the *Journal of Organizational and End User Computing* (SSCI/SCI). He is also an Associate Editor of the *Journal of Global Information Management* (SSCI/SCI) and the *Journal of Global Information Management* (SCI), and serves on the editorial boards of 20 other journals.

Md Zakirul Alam Bhuiyan (Senior Member, IEEE) worked as an Assistant Professor with Temple University. He is currently an Assistant Professor with the Department of Computer and Information Sciences, Fordham University, NY, USA, the Founding Director of the Fordham Dependable and Secure System Laboratory (DependSys). His research interests include dependability, cybersecurity, big data, and the Internet of Things/CPS applications. His work in these areas published in top-tier venues, including the IEEE TRANSACTIONS ON COMPUTERS (IEEE TC), the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (TPDS), the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC), the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS (TII), COMMAG, the IEEE INTERNET OF THINGS JOURNAL (IoT-J), *ACM Transactions on Sensor Networks* (ACM TOSN), *ACM Transactions on Autonomous and Adaptive Systems* (TAAS), CS, INF, INS, the *Journal of Scientific Agriculture* (JSA), and the *Journal of Network and Computer Applications* (JNCA). He is a member of ACM. He has been an ESI Highly Cited Researcher since 2017 and received numerous awards, including the IEEE TCSC Early Career Researcher, the IEEE Outstanding Leadership Award, and the IEEE Service Award. He has also served as an Organizer, the General Chair, the Program Chair, the Workshop Chair, and a TPC Member of various international conferences, including the IEEE INFOCOM. He has served as a Guest/Associate Editor for IEEE (TII, TBD, TCC, IoT-J), ACM (TOMM, TCPS), INS, FGCS, and JNCA.