International Journal of Innovative Computing, Information and Control
Volume **20**, Number **2**, April **2024**

ICIC International ©2024 ISSN 1349-4198
pp. 391–410

RESEARCH ON SECURE OFFLOADING OF EDGE COMPUTING FOR TRAFFIC VIDEO SURVEILLANCE

Jianhua Liu*, Xiaoni Shi, Jiajia Liu and Xiaoguang Tu

Institute of Electronic and Electrical Engineering
Civil Aviation Flight University of China
No. 46, 4th Section, Nanchang Road, Guanghan 618307, P. R. China
{ shixiaonigood; cafucljj; xguangtu }@cafuc.edu.cn
*Corresponding author: jianhuacafuc13@cafuc.edu.cn

Received June 2023; revised October 2023

Abstract. *To address the problems of how to resist internal attacks caused by offloading traffic surveillance video processing tasks to the edge cloud and how to reasonably allocate edge computing resources to reduce service latency and system energy consumption, this paper proposes a trust-based offloading scheme for edge computing tasks. Firstly, a novel trust scheme is presented, which establishes a trust feature value matrix and constructs a difference function based on linear discriminant analysis. To achieve the best classification performance, the maximization of the difference function is transformed into an optimization problem of finding the optimal weights. This scheme effectively solves the subjectivity issue in threshold determination in previous trust mechanisms. Finally, a technique for order preference by similarity to an ideal solution similarity ranking technique for multi-criteria decision making based on the entropy weight method (EW-TOM) is proposed to jointly optimize system latency and energy consumption. Simulation results show that the proposed scheme achieves more than 95% accuracy in filtering normal nodes, which is more than 20% better than other schemes. In addition, the latency and energy cost are better than other schemes, with an improvement of more than 6%.*

Keywords: Mobile edge computing, Classification, Trust, Task offloading, Entropy weight method

1. **Introduction.**

Nowadays, a large number of vehicles transmit information over the network to ensure the real-time operation of the Internet of Vehicles system. The traffic video surveillance system is a powerful guarantee for the normal operation of traffic, which can receive vehicle data consisting of location, vehicle status, and road conditions [1]. Most video surveillance systems adopt cloud-based centralized solutions. Although cloud computing provides scalable resources for accommodating data, the significant data communication overhead and latency constraints become the key bottlenecks for real-time Internet of Vehicles services. In order to achieve better real-time services, distributed solutions have been studied on video surveillance terminals [2]. Edge computing, as a distributed computing paradigm, has efficient storage resources, computing power, and network connectivity around video surveillance devices, which shortens the response time of video services [3, 4, 5, 6].

The exponential growth of network data and the widespread adoption of edge nodes present challenges for network security and user privacy protection. With the increase in traffic data and computational tasks, edge nodes are susceptible to illegal use and tampering of video data. A key issue is how to securely offload data to edge nodes [7, 8]. Therefore, in the context of edge computing, the security and responsiveness of traffic

DOI: 10.24507/ijicic.20.02.391

video surveillance play a crucial role in timely response to traffic incidents and building a better traffic environment. It is also necessary to allocate resources appropriately to minimize latency and energy consumption, thereby ensuring real-time performance [9].

To address the aforementioned problems, we propose a secure offloading scheme for mobile edge computing targeted at traffic video surveillance. This scheme utilizes a novel trust mechanism based on linear discriminant analysis (LDA) to resist internal attacks and employs the ideal solution similarity ranking technique for multi-criteria decision making based on the entropy weight method (EW-TOM) to achieve a reasonable allocation of edge computing resources, thereby achieving low latency and low energy consumption.

The main contributions of this paper are as follows.

1) Utilize machine learning algorithms to train the judgment of whether edge nodes pose a threat to communication. A novel trust scheme based on linear discriminant analysis, which is based on edge computing, is designed to classify the malicious nodes and honest nodes, effectively addressing the subjectivity issue of threshold establishment in previous trust schemes.

2) The differential evolution (DE) algorithm is employed to optimize the weight matrix, which combines the security information of edge network nodes to partition their security. This ensures the reliability of the classification results.

3) The edge nodes in the video surveillance service area are combined with trust values, delay, and energy consumption indicators. Weighting is performed using the entropy weight method to assign scores, ensuring a joint optimization of delay and energy consumption under the premise of secure data offloading.

4) The proposed scheme achieves more than 95% accuracy in filtering normal nodes, which is more than 20% better than other schemes. In addition, the latency and energy cost are better than other schemes, with an improvement of more than 6%.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 introduces and clarifies the trust classification model, time-cost model and energy consumption model in video surveillance of traffic systems. Section 4 firstly identifies the optimization objectives and then solves the multi-objective optimization problem using the EW-TOM method. Section 5 performs simulation validation in terms of security classification, delay and energy consumption, and the experimental results show that our model achieves better performance than other methods. Finally, Section 6 summarizes the work of this paper.

2. **Related Work.**

In intelligent transportation systems, the offloading of video surveillance tasks requires appropriate partitioning of computational tasks and their rational allocation to edge nodes [10, 11, 12]. The work of [4] proposes a hardware-based implementation of a joint optimization scheme for multi-user and multi-edge server computation offloading and resource allocation, aiming to reduce energy consumption while meeting task latency requirements. The work of [13] addresses the problem of low-cost task offloading in resource-constrained multi-drone edge computing environments by proposing a distributed location-aware task offloading scheme to enhance task offloading efficiency. The work of [3] introduces a deep reinforcement learning method for selecting the optimal edge server for offloading and allocating the best computational resources. These approaches primarily consider energy resource costs, but pay insufficient attention to time costs. Additionally, considering the time cost, Xu et al. present a resource allocation model [14] that adopts delay-aware scheduling and resource allocation algorithms to minimize time delay. The work of [15] proposes a task offloading strategy for heterogeneous edge systems to obtain optimal task offloading policies and reduce system latency. Wang and Yuan use a game model to reduce the energy consumption of the base station group [16].

Liu et al. optimize the delay and optimize the energy consumption by means of the knapsack problem [17]. However, the above studies mainly focus on solving problems such as time and energy delays. None of them considers security-related issues.

Edge nodes face serious challenges in privacy protection and preventing privacy leakage during task offloading. Xie et al. analyze the issue of malicious behavior by nodes during task offloading and propose a security analysis-based task offloading method to address this problem [18]. The work of [19] presents an edge computing service architecture that combines trust management methods with dynamic cost evaluation schemes to address trust-related issues. In the work of [20], an extensible trust-based security mechanism is derived to meet the security requirements of mobile edge clouds. Jiang and Tseng establish a trust mechanism for base station nodes based on mobile edge computing, providing a scientific theoretical basis and guidance for mobile edge computing applications [21]. The works of [22, 23] propose trust-aware service offloading strategies for vehicular surveillance in the context of edge computing to ensure security during data offloading. Furthermore, the work of [24] introduces the updating of trust evaluation results for mobile terminals based on environmental changes. Latif et al. also propose a trust management model that evaluates device trustworthiness based on specified weights [25]. Although security issues have been under focus, a comprehensive security decision-making system has not yet been established. Existing trust schemes typically require a threshold to distinguish between normal and malicious nodes. Unfortunately, determining an appropriate threshold for a trust scheme remains an open question. Additionally, current research efforts primarily focus on optimizing strategies to reduce user latency or energy consumption, often overlooking the trustworthiness of services.

In the above related work, only the latency can be well optimized when offloading the edge tasks is considered alone without considering the security; or considering the security but making concessions in terms of time and energy consumption. Therefore, this paper aims to achieve a comprehensive optimization of system latency and energy consumption while ensuring security. In this paper, we design a secure edge computing offloading scheme for traffic video surveillance. By training a novel secure neural network model using a large amount of node data, the scheme evaluates the security of edge nodes under different traffic scenarios. The scheme is based on the premise of secure data offloading, ensures that the nodes to which tasks are offloaded are trusted, and optimizes the operational energy consumption and computational response time of the traffic intelligent surveillance system to ensure real-time quality of service (QoS).

3. System Model.

This section presents and elucidates the trust classification model, time cost model, and energy consumption model in video surveillance in the context of transportation systems. The symbols are listed in Table 1.

3.1. Architecture for edge computing-based traffic video surveillance.

This paper applies edge computing technology to video surveillance in transportation systems. The system architecture is depicted in Figure 1, where the surveillance devices are fixedly installed at the roadside, capturing video footage as vehicles move. The mobile edge network for traffic video surveillance consists of M video surveillance devices along the roads, one cloud server, and N edge servers (each edge node is associated with one edge server, where $M > N$). The video data generated by the surveillance devices are offloaded to nearby edge nodes via wireless access points (APs). Each edge server can perform computation tasks and transmit the computation results to the cloud data center for video analysis through base stations.

Table 1. Symbol definitions

Symbols	Definitions
M	Number of video surveillance equipment
N	Number of edge nodes
P	Number of trust features
$\mathbf{I}$	Trust value
$\mathbf{T}$	Time delay
$\mathbf{E}$	Energy loss
$\mathbf{Q}$	Multiple optimization objectives
$\mathbf{w}_N$	Classification weights
$\mathbf{W}_j$	Entropy weight

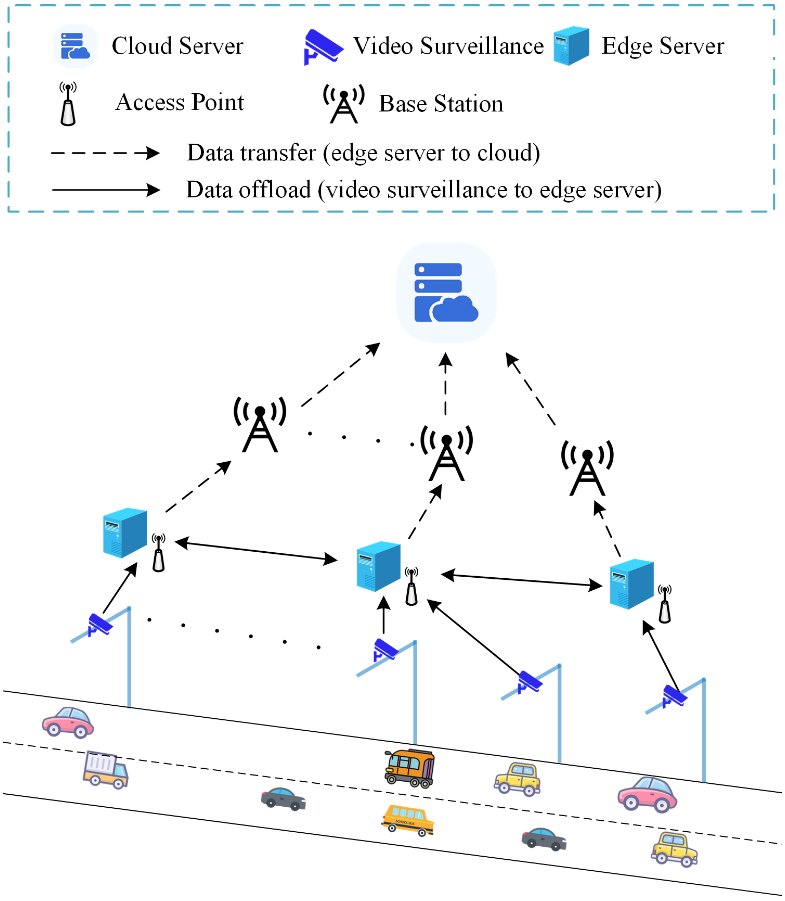

Cloud Server
Video Surveillance
Edge Server
Access Point
Base Station
Data transfer (edge server to cloud)
Data offload (video surveillance to edge server)

Figure 1. Secure offload framework for traffic video surveillance with edge computing

3.2. **Trust classification model.**

3.2.1. *Training network node information network.* The surrounding area of the traffic video surveillance system is populated with numerous edge nodes, and the trustworthiness of these edge nodes is influenced by multiple trust factors. Therefore, the establishment of the edge node trust system serves as a prerequisite and foundation for computing the trust value of edge nodes. The primary trust factors that affect the trust value of edge nodes include the number of task offloading instances received by the nodes, the average task load offloaded to the nodes, the coverage range of the nodes, the signal strength of task reception by the nodes, and the success rate of task offloading at the nodes.

To evaluate the security of edge nodes, it is necessary to collect and analyze the corresponding data of the trust feature elements of the nodes under evaluation, and establish an edge node information model. Assuming there are N edge nodes in the network and P quantized trust feature elements, the resulting matrix for quantized trust features of edge nodes $\mathbf{N}$ is as follows:

$$\mathbf{N} = \begin{pmatrix} n_{1,1} & n_{1,2} & \cdots & n_{1,P} \\ n_{2,1} & n_{2,2} & \cdots & n_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ n_{N,1} & n_{N,2} & \cdots & n_{N,P} \end{pmatrix}$$

The feature information of the edge node is denoted as $\mathbf{n}_n = (n_{n,1}, n_{n,2}, \ldots, n_{n,P})$, representing the collection of all trust quantized features of the n-th network edge node. And, n is a natural number. $n_{n,1}$ represents the first quantized feature of the n-th network edge node, while $n_{n,P}$ represents the P-th quantized feature of the n-th network edge node.

For newly joined nodes, the initial values are uniformly set to the same value within the trusted range. In subsequent offloading processes, the trustworthiness of edge nodes is reevaluated based on the unloading interaction and success rate.

Next, an information classification function needs to be established and a weight matrix designed to ensure accurate identification of trust features of edge nodes. Before establishing the information classification function, it is necessary to normalize the trust quantized feature matrix $\mathbf{N}$ of the edge nodes. Normalization is a linear transformation applied to the raw data, where the $\max\{n_{1,p}, n_{2,p}, \ldots, n_{n,p}\}$ represents the maximum value of the feature data and the $\min\{n_{1,p}, n_{2,p}, \ldots, n_{n,p}\}$ represents the minimum value of the feature data.

$$\hat{N}_{n,p} = \frac{n_{n,p} - \min\{n_{1,p}, n_{2,p}, \ldots, n_{n,p}\}}{\max\{n_{1,p}, n_{2,p}, \ldots, n_{n,p}\} - \min\{n_{1,p}, n_{2,p}, \ldots, n_{n,p}\}}, \quad \forall n \in N, \forall p \in P \tag{1}$$

For the trust quantized feature matrix $\mathbf{N}$ of edge node information, the standardized samples obtained by denoting it as $\hat{N}_{n,p}$, and then the normalized matrix is obtained $\hat{\mathbf{N}}$, the product with the weight matrix $\mathbf{w}_N$ of the quantized features is represented as $\mathbf{ZN}$, which corresponds to the target output value. It can be expressed as follows:

$$\mathbf{ZN} = \hat{\mathbf{N}} \times \mathbf{w}_N \tag{2}$$

And the mean of the trust quantized feature matrix $\mathbf{ZN}$ of the edge node is denoted by $\overline{ZN}$. The expression is as follows:

$$\overline{ZN} = \frac{1}{N}\sum_{n=1}^{N} ZN_n = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N\right), \quad \forall n \in N \tag{3}$$

where ZN_n is the value of the n-th node in matrix $\mathbf{ZN}$, and $\hat{\mathbf{N}}_n = \left(\hat{\mathbf{N}}_{n,1}, \hat{\mathbf{N}}_{n,2}, \ldots, \hat{\mathbf{N}}_{n,P}\right)$. In order to demonstrate the degree of dispersion of the distribution obtained by calculating the trust feature factors of the edge nodes based on their own trust characteristics, the covariance matrix of the information quantization features of the edge nodes is computed and denoted as $\mathbf{S}_{ZN}$. The expression is as follows:

$$\begin{aligned} \mathbf{S}_{ZN} &= \frac{1}{N}\sum_{n=1}^{N}\left(ZN_n - \overline{ZN}\right)\left(ZN_n - \overline{ZN}\right)^T \\ &= \frac{1}{N}\sum_{n=1}^{N}\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N - \overline{ZN}\right)\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N - \overline{ZN}\right)^T \end{aligned} \tag{4}$$

The covariance matrix is computed for quantifying the information quantization features of the edge nodes, which is used to characterize the similarity among different feature variables and determine their within-class dispersion. The computed results are divided into two categories: the first category represents edge nodes classified as honest nodes, and the second category represents network nodes identified as malicious nodes. For the first category of honest nodes, there are N_1 valid nodes that match with them. The sample mean is denoted as $\overline{ZN_1}$, and the covariance is denoted as S_{N_1}. The expressions for these are as follows:

$$\overline{ZN_1} = \frac{1}{N_1}\sum_{n=1}^{N_1} ZN_n = \frac{1}{N_1}\sum_{n=1}^{N_1}\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N\right) \tag{5}$$

$$S_{N_1} = \frac{1}{N_1}\sum_{n=1}^{N_1}\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N - \overline{ZN_1}\right)\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N - \overline{ZN_1}\right)^T \tag{6}$$

For the second category of nodes identified as malicious, there exist N_2 matches, with a sample mean denoted as $\overline{ZN_2}$ and a covariance denoted by S_{N_2}. The expressions are as follows:

$$\overline{ZN_2} = \frac{1}{N_2}\sum_{n=1}^{N_2} ZN_n = \frac{1}{N_2}\sum_{n=1}^{N_2}\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N\right) \tag{7}$$

$$S_{N_2} = \frac{1}{N_2}\sum_{n=1}^{N_2}\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N - \overline{ZN_2}\right)\left(\hat{\mathbf{N}}_n \times \mathbf{w}_N - \overline{ZN_2}\right)^T \tag{8}$$

In order to obtain a robust security model, the linear discriminant analysis (LDA) approach [13] is employed, ensuring small within-class variations and large between-class separations, thus achieving high cohesion and low coupling. By continuously training the classification weights, the high-dimensional data belonging to the same class are projected onto a lower-dimensional space, bringing together samples of the same category while keeping a considerable distance from different categories. Ultimately, the trust quantization features of the high-dimensional edge nodes are projected onto a one-dimensional line. To enhance the algorithm's adaptability and flexibility, an improved algorithm called DE/rand2best/1/bin is proposed, where rand2best always compares with the historical best solution. The number of difference vectors used is one, and a binary bin crossover operation method is employed to assess the security of network node information [14].

In the security classification model, the inter-class difference is represented by $\left(\overline{ZN_1} - \overline{ZN_2}\right)^2$, and a greater inter-class difference is desired. The intra-class difference is represented by $S_{N_1} + S_{N_2}$, and a smaller intra-class difference is desired. Therefore, a function model $F_{\mathbf{N}_i}$ is established for the classification model to satisfy the aforementioned assumptions, where the inter-class difference is directly proportional to $F_{\mathbf{N}_i}$ and the intra-class difference is inversely proportional to $F_{\mathbf{N}_i}$. The expression is as follows:

$$F_{\mathbf{N}_i}(\mathbf{w}_N) = \frac{\left(\overline{ZN_1} - \overline{ZN_2}\right)^2}{S_{N_1} + S_{N_2}} \tag{9}$$

The problem of the classification model is thus transformed into finding the value of $\mathbf{w}_N$ when $F_{\mathbf{N}_i}$ is maximized, which can be expressed using the following equation:

$$\hat{\mathbf{w}}_N = \arg\max_{\mathbf{w}_N} F_{\mathbf{N}_i}(\mathbf{w}_N), \quad \hat{\mathbf{w}}_N \in (0,1) \tag{10}$$

3.2.2. *Optimization of the weight vector.* In the context of security classification problems, the aforementioned problem has been transformed into a function model of weights. In order to obtain a robust security classification model, the differential evolution (DE) algorithm is utilized for weight optimization. The steps for optimizing the aforementioned weights of network information quantization features $\mathbf{w}_N$, are as follows.

1) Initializing individuals: The differential evolution algorithm utilizes N_p real-valued parameter vectors of dimension D as the population for each generation, with each individual represented as $\mathbf{w}_{i,G}$ $(i = 1, 2, \ldots, N_p)$, where i denotes the sequence of individuals in the population, G denotes the evolution generation, and N_p denotes the population size. Assuming that all randomly initialized populations follow a uniform distribution, the limits for the quantization feature weights are set as follows:

$$\mathbf{w}_N^{(L)} \leq \mathbf{w}_N \leq \mathbf{w}_N^{(U)} \tag{11}$$

In the above equation, $\mathbf{w}_N^{(L)}$ and $\mathbf{w}_N^{(U)}$ respectively represent the lower and upper bounds of the quantization feature weights $\mathbf{w}_N$, yielding the following relationship:

$$\mathbf{w}_{N_i,0} = \text{rand}[0,1]\left(\mathbf{w}_N^{(U)} - \mathbf{w}_N^{(L)}\right) + \mathbf{w}_N^{(U)}, \quad i = (1, 2, \ldots, N_p) \tag{12}$$

$\mathbf{w}_{N_i,0}$ represents the initial value of the i-th quantization feature weight, and rand$[0,1]$ represents a uniformly random number selected from the range of 0 to 1.

2) Mutation operation: Adaptive mutation is performed using a random-to-best mutation method to mutate towards the direction of the best quantization feature weights. The expression is as follows:

$$\mathbf{v}_{i,G+1} = \mathbf{w}_{i,G+1} + \lambda\left(\mathbf{w}_{best,G} - \mathbf{w}_{i,G}\right) + M\left(\mathbf{w}_{r_1,G} - \mathbf{w}_{r_2,G}\right) \tag{13}$$

In the above equation, $\mathbf{v}_{i,G+1}$ represents the newly generated individual after mutation, $\mathbf{w}_{i,G}$ represents the current quantization feature weights, $\mathbf{w}_{best,G}$ represents the best historical quantization feature weights, r_1 and r_2 denote randomly selected indices. It is evident that $r_1 \neq r_2 \neq i$ and λ are adaptive mutation operators, and M is the traditional mutation operator, which involves randomly selecting two different individuals from the population, scaling their vector difference, and combining it with the individual to be mutated. The λ satisfies the following relationship:

$$\lambda = \exp\left(1 - \frac{G_m}{G_m + 1 - G}\right) \tag{14}$$

where, G_m represents the maximum evolution generation, G represents the current evolution generation. M satisfies $M = M_0 2^{\lambda}$. M_0 is the initial mutation coefficient, which ranges between 0 and 2 and controls the scaling of the bias variable. A smaller M value enhances the local search capability of the DE algorithm, while a larger M value allows the DE algorithm to escape local optima and search for global optimal solutions. Thus, the differential evolution algorithm possesses adaptive mutation capability.

3) Crossover operation: To increase the diversity of interfering parameter vectors, a crossover operation is employed:

$$\mathbf{w}_{i,G+1} = \left(w_{1i,G+1}, w_{2i,G+1}, \ldots, w_{Di,G+1}\right) \tag{15}$$

$$\mathbf{w}_{ki,G+1} = \begin{cases} \mathbf{v}_{ki,G+1}, & \text{if } \text{rand}(k) \leq C \text{ or } k = R(i) \\ \mathbf{w}_{ki,G+1}, & \text{if } \text{rand}(k) > C \text{ or } k \neq R(i) \end{cases} \tag{16}$$

For D-dimensional quantization feature weights, they can be represented by the vector $\mathbf{w}_{i,G+1}$. In the equation, rand(k) denotes a random number drawn from a uniform distribution within $[0,1]$, C is the crossover coefficient, and $R(i) \in \{1, 2, \ldots, D\}$. If the random number is less than or equal to the crossover coefficient, or if the dimension matches the randomly selected dimension, the quantization feature weight result will undergo crossover with the mutation result; otherwise, it remains unchanged. To ensure that at least one parameter from the intermediate mutated vector $\mathbf{v}_{ki,G+1}$ is passed on to the next generation, the first gene of the crossover operation is randomly chosen as one of the parameters of $\mathbf{w}_{ki,G+1}$ at the same position after crossover. Subsequent crossover operations select the parameters at the same position based on the crossover coefficient C. The crossover operation of the algorithm helps to introduce diversity, which helps to explore the breadth of the solution space by combining information from individuals to produce new individuals. This diversity helps to prevent the algorithm from falling into local optimal solutions.

The C parameter controls the weights of the vector differences, and the choice of the C parameter directly affects the performance of the DE algorithm. Different values of C will lead to different behaviours of the algorithm, if the value of C is small, this may cause the algorithm to fall into local optimal solutions; if the value of C is large, this may increase the diversity of the algorithm and reduce the convergence speed of the algorithm. Therefore, a reasonable choice of C value is important to optimize the DE algorithm, and, Equations (15) and (16) are reasonable.

To introduce more dynamics and increase the persuasiveness of the obtained results, C is subjected to random transformation. The crossover operator C, with a random range, is described by the following equation:

$$C = 0.5 * (1 + \mathrm{rand}[0, 1]) \tag{17}$$

4) Selection operation: In the differential evolution algorithm, the new individuals are compared with the target individuals in the current population based on a greedy criterion. The individual with a superior fitness value between the two will be selected to enter the next generation.

$$\mathbf{w}_{i,G+1} = \begin{cases} \mathbf{v}_{i,G+1}, \text{ if } F\left(\mathbf{v}_{i,G+1}\right) \geq F\left(\mathbf{w}_{i,G+1}\right) \\ \mathbf{w}_{i,G+1}, \text{ if } F\left(\mathbf{v}_{i,G+1}\right) < F\left(\mathbf{w}_{i,G+1}\right) \end{cases} \tag{18}$$

In the above equation, F represents the objective function, and a larger value of the objective function indicates a higher reliability of the classification based on the weight $\mathbf{w}_{i,G+1}$.

5) Boundary condition handling: Due to the existence of upper and lower bounds for the weights in this instance, it is necessary to ensure that the parameter values of the newly generated individuals are within the feasible domain of the problem. Any newly generated individual that does not satisfy the boundary constraints will be replaced with a parameter vector randomly generated within the feasible domain. Therefore, we have

$$\mathbf{w}_{N_i,G+1} = \mathrm{rand}[0, 1]\left(\mathbf{w}_N^{(U)} - \mathbf{w}_N^{(L)}\right) + \mathbf{w}_N^{(L)} \tag{19}$$

The final algorithm obtains the strongest fitness $\mathbf{w}_N$ as the weights for training the neural network, denoted as $\hat{\mathbf{w}}_N = \mathbf{w}_N$. The neural network model trained according to this criterion exhibits the best classification performance in this scenario. Additionally, the trust value of the marginal nodes can be calculated based on these weights.

$$\mathbf{I} = \hat{\mathbf{N}} \times \hat{\mathbf{w}}_N, \quad \mathrm{I}_i \in [0, 1] \tag{20}$$

In the above equation, $\mathbf{I}$ represents the trust value matrix of the marginal nodes, and I_i represents the trust value of the i-th node. Therefore, during the task offloading process, it is easy to obtain the trust value of offloading to the marginal nodes.

3.2.3. *Edge offloading security judgment.* During the information collection and exchange process, real-time information from video monitoring is collected. Under the optimal trust quantization feature weights, the input is fed into the traffic edge network model to determine the security of user real-time access to edge nodes. This can be divided into two scenarios: if a user accesses a malicious edge node (classified by the model as an untrusted node or having a trust value below a certain threshold), the malicious edge node will be blacklisted, while regular users will continue to queue and wait to establish communication with other edge nodes; if a user accesses a normal edge node, the communication connection proceeds normally.

3.3. **Time cost model.** In a traffic video surveillance system, latency is one of the important factors that affect traffic and must be considered when optimizing the delay caused by communication between wireless devices and edge servers. Time cost includes the migration time for offloading and the computation time of the corresponding edge nodes. As a critical parameter, the time cost determines the quality of real-time services in the surveillance devices.

The quantification of the uplink transmission rate between video surveillance and servers is given as follows:

$$C_n = B_n \log_2 \left(1 + \frac{P_i h_n(t)}{w_0}\right) \tag{21}$$

In the above equation, B_n represents the bandwidth of the uplink transmission link, w_0 represents the power of white noise, P_i represents the transmission power of video surveillance when offloading its computation tasks to edge servers, and $h_n(t)$ represents the corresponding channel gain, assuming that this value remains constant during the transmission. The migration time t_1 for offloading computation tasks can be calculated as follows:

$$t_1 = \sum_{i=1}^{n} \frac{\alpha_i}{C_n} \tag{22}$$

where α_i is the calculated unloading task volume.

For the computation time at the edge server, f_n represents the CPU cycles of the edge server (unit: cycles/second), and β_i represents the number of CPU cycles required to complete the computation task. Therefore, the computation latency t_2 at the edge node is given by

$$t_2 = \sum_{i=1}^{n} \frac{\beta_i}{f_n} \tag{23}$$

Thus, the total offload delay T is obtained:

$$T = t_1 + t_2 \tag{24}$$

3.4. **Energy cost model.** Energy consumption is also a crucial factor in traffic video surveillance. It is important to ensure both high trust values and low latency while minimizing energy costs. The energy optimization in a traffic video surveillance system mainly consists of two parts: the energy consumed in uploading surveillance data to edge nodes and the energy consumed in processing the data at the edge nodes.

Firstly, the energy consumption when video surveillance i offloads a computation task to edge node n is given by

$$E_{in}^{tran} = \sum_{i=1}^{M} \left(Power_i \times t_1\right) \tag{25}$$

where $Power_i$ denotes the power consumption of the i-th video surveillance device for task offloading transmission. Next, the energy consumption for processing data at the edge node is given by

$$E_{in}^{cal} = \sum_{i=1}^{N} (Power_n \times t_2) \tag{26}$$

where $Power_n$ represents the computing power of the n-th edge server.

The total energy consumption for offloading is obtained as follows:

$$E = E_{in}^{tran} + E_{in}^{cal} \tag{27}$$

In contrast to other works, this paper combines reliability and delay energy optimization. Firstly, machine learning algorithms are used to train to determine whether edge nodes pose a threat to communication. A novel trust scheme based on linear discriminant analysis is designed to classify malicious nodes and honest nodes, which effectively solves the subjectivity problem of threshold establishment in previous trust schemes. The differential evolution (DE) algorithm is used to optimize the weight matrix and classify the edge network nodes as honest nodes by combining their trust characteristics. It ensures the reliability of the classification results. Secondly, the trust value, delay and energy consumption metrics of the edge nodes in the video surveillance service area are jointly optimized. In the following, entropy weighting method will be used for weighted assignment to ensure the joint optimization of delay and energy consumption under the premise of data security offloading.

4. Design of the EW-TOM Rating Model.

In this section, the optimization objective is first identified. Then, the EW-TOM method is used to solve the multi-objective optimization problem. Entropy weighting and multi-criteria decision-making methods are used to help decision makers select the best option under multiple evaluation metrics or decision criteria. The entropy weighting method usually does not require the decision maker to provide subjective weights, as it can automatically calculate weights based on data. The entropy weighting method is more objective and applicable to problems with multiple metrics or decision criteria, as it automatically assigns weights to each metric. Multi-criteria decision-making methods can be tailored to meet the needs of a wide range of situations, depending on the nature of the problem and on how the distance metrics are used. Their fusion results in the EW-TOM method, which is suitable for the selection of offloading options.

In this paper, the objectives include minimizing time overhead and minimizing energy consumption under the condition of secure offloading. The task offloading problem is formulated as a multi-objective optimization problem. The problem formulation is represented as follows:

$$\min \mathbf{Q} = \{\min T, \min E\}, \quad I \geq I_{\min} \tag{28}$$

The constraint represents that the edge nodes must be honest nodes.

Based on the trust classification model in Section 3, it can be determined whether an edge node is trustworthy. When accessing a malicious node, the connection can be interrupted and the task offloading can be aborted. However, in the entire network, there are other secure and trustworthy nodes that can provide the same services. Therefore, it is crucial to establish a task offloading scheme that ensures security while considering low latency and low energy consumption metrics. In this subsection, the ideal solution similarity ranking technique for multi-criteria decision making based on the entropy weight method (EW-TOM) is used to score the security nodes within the user service requirements, aiming to find the optimal offloading solution.

4.1. **Entropy method to calculate information utility value.** The problem matrix $\mathbf{Q}$ eliminates the effect of different indicator magnitudes and normalizes to obtain the matrix $\widehat{\mathbf{Q}}$:

$$\widehat{\mathbf{Q}} = \begin{pmatrix} \widehat{Q}_{1,1} & \widehat{Q}_{1,2} & \cdots & \widehat{Q}_{1,m} \\ \widehat{Q}_{2,1} & \widehat{Q}_{2,2} & \cdots & \widehat{Q}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{Q}_{n,1} & \widehat{Q}_{n,2} & \cdots & \widehat{Q}_{n,m} \end{pmatrix}$$

Using the entropy weight method, the weight of the i-th offloading scheme under the j-th criterion is calculated, and it is considered as the probability used in the calculation of relative entropy. In order to obtain the probabilities used in the calculation of relative entropy, the probability matrix $\mathbf{P}$ is obtained, where the calculation formula for each element $p_{i,j}$ of $\mathbf{P}$ is as follows:

$$p_{i,j} = \frac{\widehat{Q}_{i,j}}{\sum_{i=1}^{n} \widehat{Q}_{i,j}} \tag{29}$$

where $\widehat{Q}_{i,j}$ is every offloading strategy. Moreover, $\sum_{i=1}^{n} p_{i,j} = 1$. Then, the information entropy of each indicator is calculated and the information utility value is calculated and normalized to obtain the entropy weight of each indicator. For the j-th criterion, the formula for calculating the information entropy is

$$h_j = -\frac{\sum_{i=1}^{n} p_{i,j} \ln (p_{i,j})}{\ln n}, \quad (j = 1, 2, \ldots, m) \tag{30}$$

where h_j represents the information entropy of the j-th criterion. A larger information entropy indicates less information for the j-th criterion. When $p_{1,j} = p_{2,j} = \cdots = p_{n,j}$, the information entropy reaches its maximum value, which is $h_j = 1$. To normalize it, let T_j represent the information utility value of the j-th criterion, given by the following expression:

$$T_j = 1 - h_j \tag{31}$$

That is, the higher the information utility value, the more information it contains. By normalizing the information utility values, we can obtain the entropy weight for each criterion. Based on the entropy weights, we can allocate weights to the delay and energy consumption. The entropy weight is calculated as follows:

$$\mathbf{W}_j = \frac{T_j}{\sum_{j=1}^{m} T_j} \tag{32}$$

4.2. **TOM scoring algorithm.** Based on the objective scoring of each criterion mentioned above [26], we can calculate the ideal solution as follows:

$$\begin{aligned} \widehat{\mathbf{Q}}^* &= \left(\widehat{\mathbf{Q}}_1^*, \widehat{\mathbf{Q}}_2^*, \ldots, \widehat{\mathbf{Q}}_m^*\right) \\ &= \left(W_1 \cdot \max\left\{\widehat{Q}_{1,1}, \widehat{Q}_{2,1}, \ldots, \widehat{Q}_{n,1}\right\}, W_2 \cdot \max\left\{\widehat{Q}_{1,2}, \widehat{Q}_{2,2}, \ldots, \widehat{Q}_{n,2}\right\}, \ldots,\right. \\ &\quad \left. W_m \cdot \max\left\{\widehat{Q}_{1,m}, \widehat{Q}_{2,m}, \ldots, \widehat{Q}_{n,m}\right\}\right) \end{aligned} \tag{33}$$

The non-ideal solution can be represented as follows, where $\widehat{\mathbf{Q}}_1^*$ is the maximum value of the weighted entropy for the first criterion, and $\widehat{\mathbf{Q}}_m^*$ is the maximum value of the weighted entropy for the m-th criterion.

$$\begin{aligned}\widehat{\mathbf{Q}}^- &= \left(\widehat{\mathbf{Q}}_1^-, \widehat{\mathbf{Q}}_2^-, \ldots, \widehat{\mathbf{Q}}_m^-\right) \\ &= \left(W_1 \cdot \min\left\{\widehat{Q}_{1,1}, \widehat{Q}_{2,1}, \ldots, \widehat{Q}_{n,1}\right\}, W_2 \cdot \min\left\{\widehat{Q}_{1,2}, \widehat{Q}_{2,2}, \ldots, \widehat{Q}_{n,2}\right\}, \ldots, \right. \\ &= \left. W_m \cdot \min\left\{\widehat{Q}_{1,m}, \widehat{Q}_{2,m}, \ldots, \widehat{Q}_{n,m}\right\}\right) \end{aligned} \tag{34}$$

where $\widehat{\mathbf{Q}}_1^-$ is the minimum value of the weighted entropy for the first criterion, and $\widehat{\mathbf{Q}}_m^-$ is the minimum value of the weighted entropy for the m-th criterion.

The weights have been applied to $\mathbf{Q}^*$ and $\mathbf{Q}^-$ before calculating G_i^* and G_i^-, and the $\mathbf{Q}^*$ and $\mathbf{Q}^-$ in the formula are already new data after combining with the weights. Therefore, the weights are equal. The distance G_i^* between the i-th object and the ideal solution can be defined as follows:

$$G_i^* = \sqrt{\sum_{j=1}^{m} \left(\widehat{\mathbf{Q}}_j^* - \widehat{\mathbf{Q}}_{i,j}^*\right)^2} \tag{35}$$

Similarly, the distance G_i^- between the i-th object and the non-ideal solution can be defined as follows:

$$G_i^- = \sqrt{\sum_{j=1}^{m} \left(\widehat{\mathbf{Q}}_j^- - \widehat{\mathbf{Q}}_{i,j}^-\right)^2} \tag{36}$$

Finally, the score of the i-th object can be calculated using the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) scoring formula.

$$F_i = \frac{G_i^-}{G_i^- + G_i^*} \tag{37}$$

The EW-TOM method combines the advantages of entropy weight method, multi-criteria decision-making, and distance-based ranking of superior and inferior solutions. Compared to traditional algorithms, it can provide secure allocation solutions for both types of indicators (max-type and min-type) and various user service requirements. The pseudocode for selecting the optimal solution is presented in Algorithm 1.

Algorithm 1: Selecting the optimal strategy

```
Output: $\mathbf{Q}$, $\mathbf{I}$, $\mathbf{T}$, $\mathbf{E}$
Ensure: $F_i$
for $i = 1 : N$ do
    Calculate $h_j$, $T_j$, $\mathbf{W}_j$
    Determine the ideal and negative-ideal solution
    Calculate the alternative separation for the ideal and negative-ideal solution
      respectively
    Calculate the relative closeness of $F_i$
end for
return $F_i$
```

5. **Experimental Simulation and Result Evaluation.** In this section, we analyze the proposed secure classification model through simulations. Under the secure classification model, we perform task offloading and use the EW-TOM method for scoring to identify the optimal offloading strategy. The simulations are conducted in MATLAB 2020b on a PC equipped with an AMD Ryzen 7 5800H with Radeon Graphics processor and 16.0 GB RAM. To ensure the validity of the experiments, we compare the EW-TOM algorithm with other algorithms using the real dataset from video surveillance nodes in Nanjing City [26], as shown in Figure 2. The basic parameters and their ranges are listed in Table 2 to provide a comprehensive understanding of the experiments.

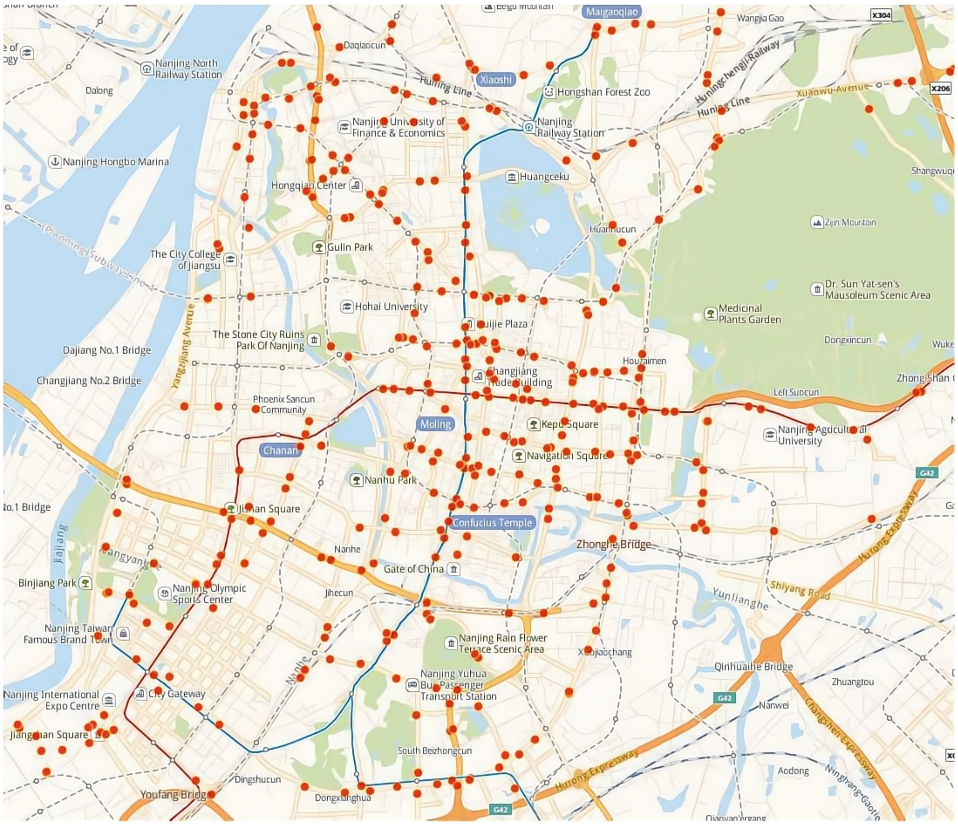


FIGURE 2. Location distribution map of video surveillance nodes in Nanjing

TABLE 2. System parameters

Parameters	Numerical value/value range
N	500
Percentage of malicious nodes	$[5\%, 30\%]$
Node feature category P	4
Category	2
G	500
B_n	10 MHz
w_0	10^{-9} W/Hz
$Power_i$	0.2 W
Number of offloading edge nodes	$[10, 50]$
Number of offloading tasks	$[50, 150]$
Task data volume size	$[1, 4]$ MB

In the process of task offloading, the density of edge node distribution directly affects the effectiveness of task offloading. A dense distribution can reduce latency and minimize energy consumption, while a sparse distribution may have the opposite effect. As shown in Figure 3, under the condition of fixed task size and quantity, the overall optimization

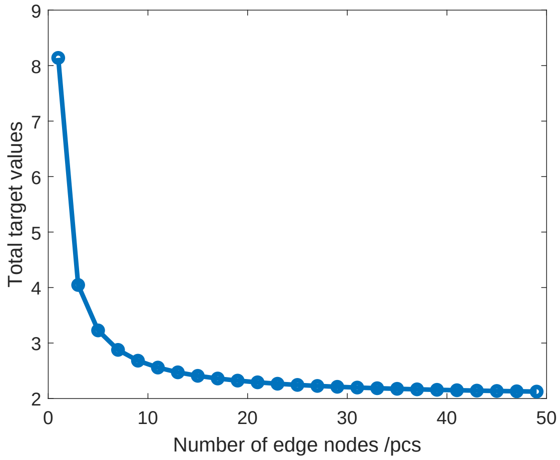
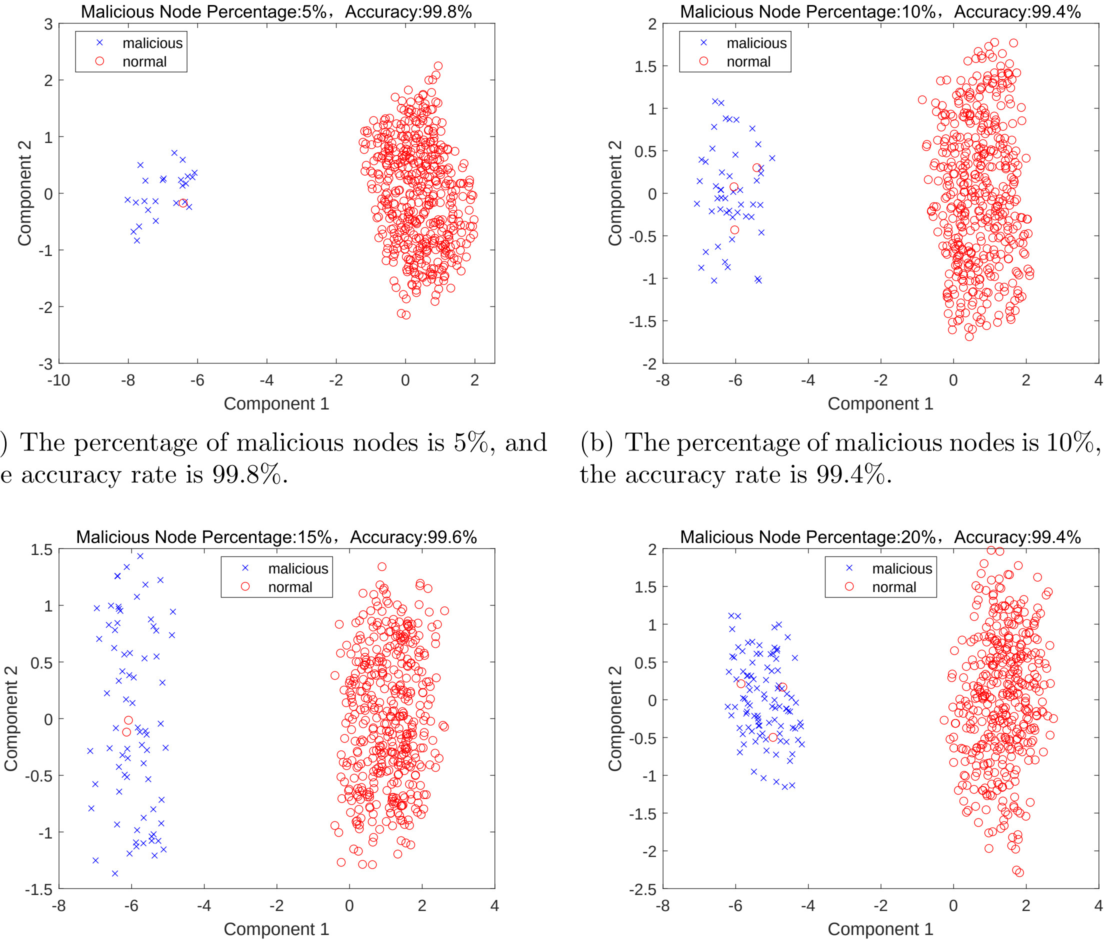


FIGURE 3. Impact of edge node distribution on task offloading



(a) The percentage of malicious nodes is 5%, and the accuracy rate is 99.8%.

(b) The percentage of malicious nodes is 10%, and the accuracy rate is 99.4%.

(c) The percentage of malicious nodes is 15%, and the accuracy rate is 99.6%.

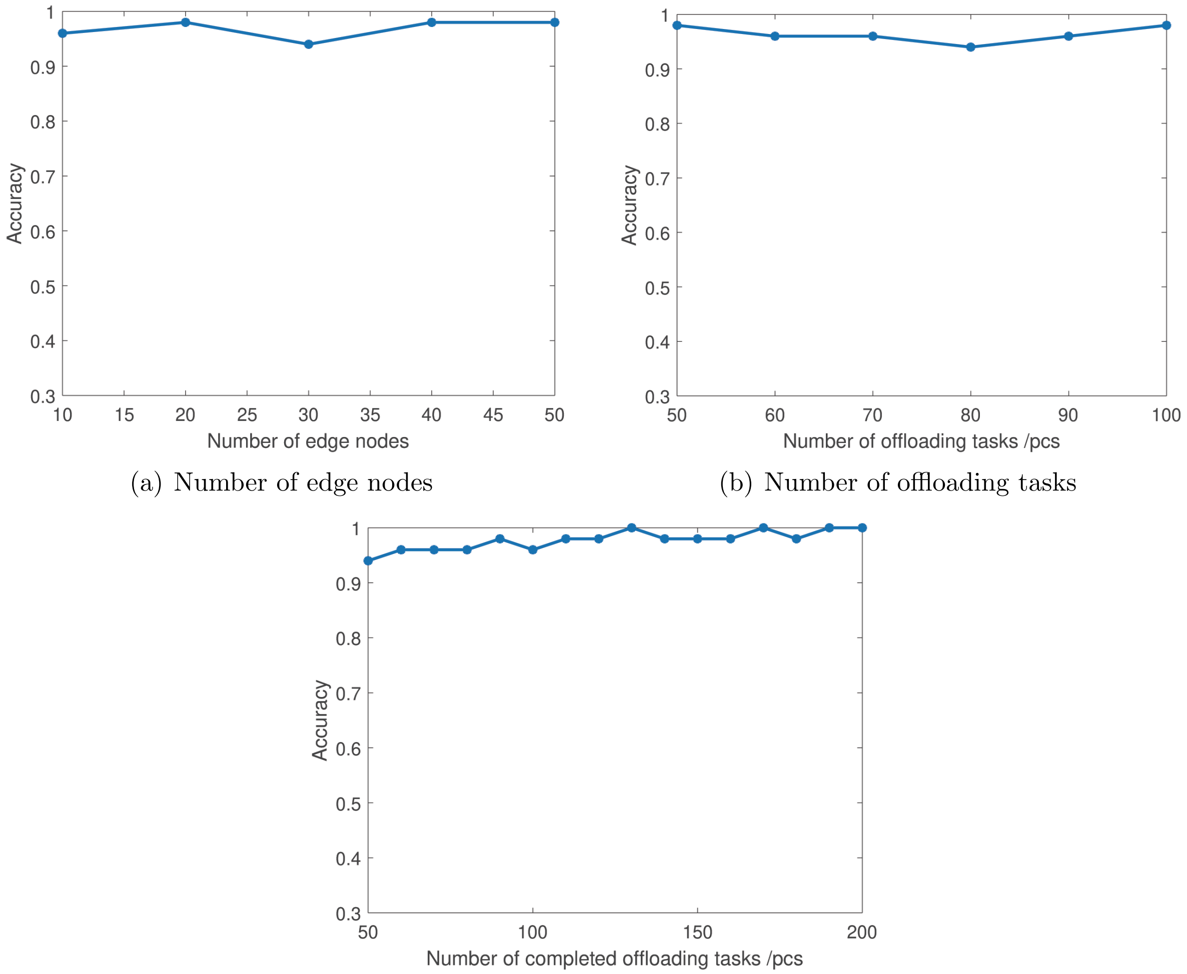
(d) The percentage of malicious nodes is 20%, and the accuracy rate is 99.4%.

FIGURE 4. The performance of our trust classification algorithm

objective of task offloading decreases as the number of edge nodes increases and tends to stabilize. Therefore, based on the Figure 3, we set the number of edge nodes to be between 10 and 50 to achieve stable performance and effectiveness in optimizing task offloading.

The training of the secure classification model is conducted in an environment with a total number of $N = 500$ edge nodes. Under different proportions of malicious secure nodes, the accuracy of the trained security classification model may vary. Through MATLAB simulations, we experimentally obtained the accuracy of the model under different proportions of malicious nodes ranging from 5% to 20%, as shown in Figure 4. It can be observed that the accuracy of the classification model exceeds 96%.

The classification accuracy of the trust classification model is shown in Figure 5. From Figure 5(a), it can be visually observed that as the number of edge nodes varies in the edge offloading environment, the probability of accurately classifying the edge nodes remains above 95%. When the number of edge nodes remains constant and the number of offloaded tasks changes, the accuracy of the classification model still performs well, staying above 96% as shown in Figure 5(b). With an increasing number of completed offloaded tasks in the environment, the historical evaluation data for the edge nodes becomes more comprehensive, leading to an excellent classification of edge node trustworthiness, with an accuracy exceeding 96% as shown in Figure 5(c).



(a) Number of edge nodes

(b) Number of offloading tasks

(c) Number of completed offloading tasks

Figure 5. Precision comparison of trust solutions

In Figure 6, we compared our proposed solution with two other approaches [27, 28] to confirm the stability and efficiency of our model. From Figure 6, it is evident that our approach exhibits higher accuracy, significantly surpassing the other two solutions, with a task offloading ratio to trusted nodes exceeding 95%. While the solution in the work of [27] shows an increasing trend, its accuracy still falls short of our approach and there is a considerable gap. The solution in the work of [28] achieves an accuracy of approximately 70%, much lower than the accuracy of our approach. With our classification scheme, the likelihood of offloading tasks to trusted edge nodes can reach a higher level. And compared with ITCN, the accuracy of security classification in this paper improves about 37% accuracy; compared with group-agent strategy, the accuracy of security classification in this paper improves about 20% accuracy. This will greatly improve the probability of task offloading to reliable nodes.

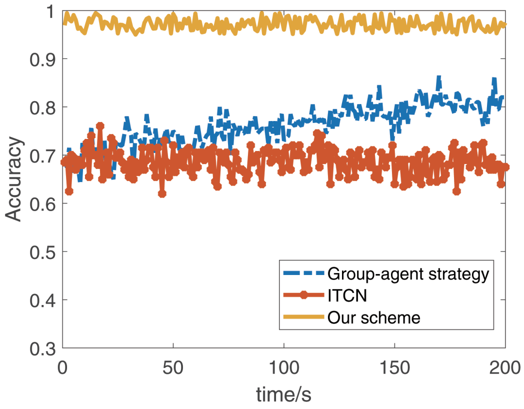


FIGURE 6. Accuracy comparison of trust solutions

In the classification model algorithm, the trust values when offloading edge computing tasks to secure and malicious nodes are shown in Figure 7. The trust values for secure nodes converge around 0.65, while the trust values for malicious nodes converge around 0.22. The security of the edge nodes can be clearly distinguished, and there is a significant difference between the two categories. Based on this model, the edge nodes can be securely classified, and a reliable threshold can be set to differentiate between the nodes.

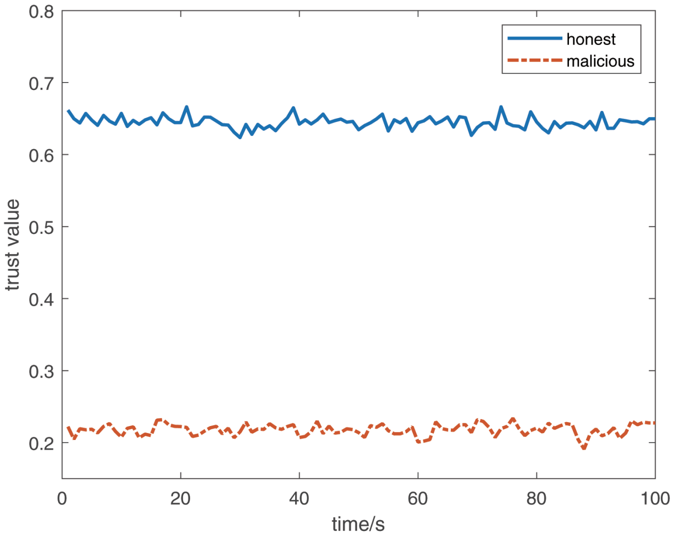


FIGURE 7. Classification trust value

Considering the substantial difference in trust values between the two classes, a threshold of 0.4 can be set in subsequent experiments.

According to the objective function (28), which minimizes the joint value of latency and energy consumption, Figure 8 clearly shows that under the same task load, the EW-TOM method outperforms BA (bat algorithm) [29], GA (genetic algorithm) [30], and the conventional random allocation algorithm in terms of the objective value, i.e., achieving the minimum joint value of latency and energy consumption. This means that the EW-TOM method can efficiently handle real-time scenarios with lower energy consumption in shorter time frames. Furthermore, as the task load increases, it can be observed that the EW-TOM algorithm consistently outperforms the other three algorithms, ensuring security while providing better services in a shorter time and reducing system energy consumption. And compared to other algorithms, the performance of EW-TOM improves on average by more than 13% as the volumes of tasks increase.

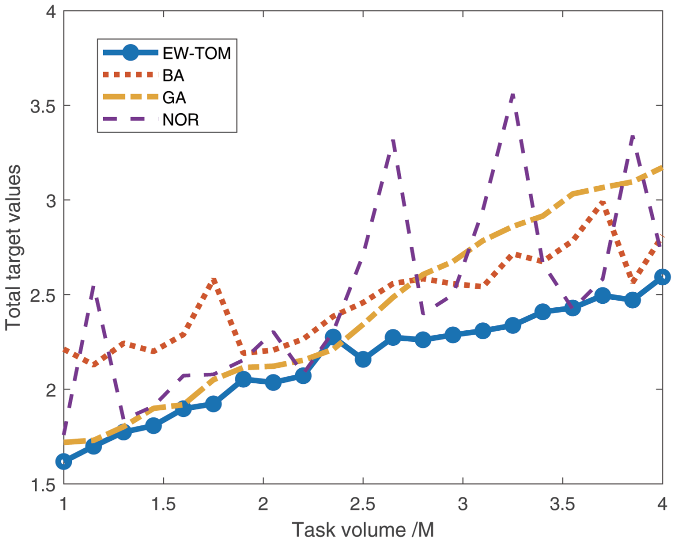


Figure 8. Variation of total target value with task volume

As shown in Figure 9, with an increase in CPU cycle count, the processor's data processing capability enhances, leading to a decrease in the overall objective value of latency and energy consumption. Even with an increasing CPU cycle count, our EW-TOM method remains significantly superior to the other three methods, ensuring the lowest overall value of latency and energy consumption and achieving energy efficiency with shorter task processing time. Figure 9 also indicates that the normal random (NOR) method exhibits larger fluctuations compared to the other methods, indicating suboptimal performance. With the increased computational power of the edge server, the overall optimization of latency-energy consumption of EW-TOM improves by more than 6% compared to other algorithms.

6. **Conclusion.** In order to address the issue of internal attacks caused by offloading traffic surveillance video processing tasks to the edge cloud, as well as the problem of rational allocation of edge computing resources to reduce service latency and system energy consumption, this paper proposes a secure task offloading scheme for traffic video surveillance based on edge computing. Firstly, a novel trust scheme based on LDA (linear discriminant analysis) is introduced. This scheme addresses the subjectivity issue in threshold setting of trust mechanisms by conducting multiple training sessions on objective data. Simulation results demonstrate that the accuracy of selecting normal nodes using our scheme

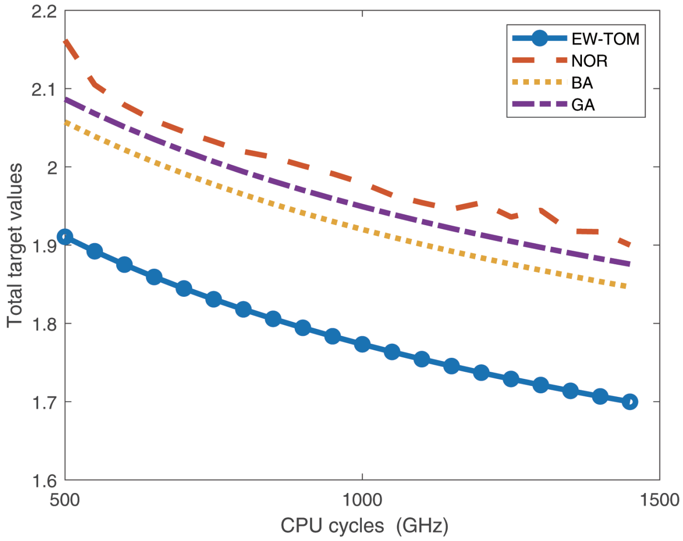


Figure 9. Variation of the total target value with the number of CPU cycles

exceeds 95%, which is significantly higher than other similar approaches. After selecting the normal nodes, the joint optimization of system latency and energy consumption is performed using the EW-TOM method. Our solution ensures the minimal latency and energy consumption in the secure offloading environment of the traffic network, providing real-time information to users in the transportation system for better emergency handling. It optimizes the operational energy consumption of the traffic intelligent monitoring system and the response time of computational tasks, ensuring the optimization of latency and energy consumption costs in edge computing services. Simulation results show that the proposed scheme achieves more than 95% accuracy in filtering normal nodes, which is more than 20% better than other schemes. In addition, the latency and energy cost are better than other schemes, with an improvement of more than 6%.

Acknowledgment. This work is partially supported by the Fundamental Research Funds for the Central Universities (J2023-027). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] V. Petrov, S. Andreev, M. Gerla et al., Breaking the limits in urban video monitoring: Massive crowd sourced surveillance over vehicles, *IEEE Wireless Communications*, vol.25, no.5, pp.104-112, 2018.

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

[3] L. Ale, N. Zhang, X. Fang et al., Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning, *IEEE Transactions on Cognitive Communications and Networking*, vol.7, no.3, pp.881-892, 2021.

[4] J. L. Zhi, N. Wang, Y. Man et al., Hardware-aware offloading and resource allocation for edge computing, *Journal of Beijing University of Posts and Telecommunications*, vol.45, no.2, pp.22-28, 2022.

[5] K. Cao, Y. Liu, G. Meng et al., An overview on edge computing research, *IEEE Access*, pp.85714-85728, 2020.

[6] Y. Xiao, Y. Jia, C. Liu et al., Edge computing security: State of the art and challenges, *Proc. of the IEEE*, vol.107, no.8, pp.1608-1631, 2019.

[7] B. H. Husain and S. Askar, Survey on edge computing security, *International Journal of Science and Business*, vol.5, no.3, pp.52-60, 2021.

[8] Y. Guo, F. Liu, N. Xiao et al., Task-based resource allocation bid in edge computing micro datacenter, *Comput. Mater. Contin.*, vol.61, pp.777-792, 2019.

[9] X. Xue, H. Han, S. Wang et al., Computational experiment-based evaluation on context-aware O2O service recommendation, *IEEE Transactions on Services Computing*, vol.12, no.6, pp.910-924, 2016.

[10] J. Zhao, Q. Li, Y. Gong et al., Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks, *IEEE Transactions on Vehicular Technology*, vol.68, no.8, pp.7944-7956, 2019.

[11] X. Li, L. Zhao, K. Yu et al., A cooperative resource allocation model for IoT applications in mobile edge computing, *Computer Communications*, vol.173, pp.183-191, 2021.

[12] G. Zhang, S. Zhang, W. Zhang et al., Joint service caching, computation offloading and resource allocation in mobile edge computing systems, *IEEE Transactions on Wireless Communications*, vol.20, no.8, pp.5288-5300, 2021.

[13] J. Liu, Z. Wu, J. Liu et al., Distributed location-aware task offloading in multi-UAVs enabled edge computing, *IEEE Access*, vol.10, pp.72416-72428, 2022.

[14] J. Xu, B. Palanisamy, H. Ludwig et al., Zenith: Utility-aware resource allocation for edge computing, *IEEE International Conference on Edge Computing (EDGE)*, pp.47-54, 2017.

[15] W. Li and S. Jin, Performance evaluation and optimization of a task offloading strategy on the mobile edge computing with edge heterogeneity, *The Journal of Supercomputing*, vol.77, no.11, pp.12486-12507, 2021.

[16] L. Wang and Z. Yuan, Efficient task offloading strategy for low-energy base station groups in mobile edge computing, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1531-1548, 2021.

[17] J. Liu, Z. Wu, J. Shen, J. Liu and X. Tu, Artificial potential field-based resource allocation for mobile edge computing, *International Journal of Innovative Computing, Information and Control*, vol.18, no.5, pp.1413-1429, 2022.

[18] N. Xie, W. A. Tan, Y. Cao et al., Modeling and analysis of secure information flow in mobile edge computing, *Computer Engineering*, vol.48, no.5, pp.35-42+52, 2022.

[19] J. Liu and Z. Wu, PECSA: Practical edge computing service architecture applicable to adaptive IoT-based applications, *Future Internet*, vol.13, no.11, 294, 2021.

[20] F. N. Nwebonyi, R. Martins and M. E. Correia, Reputation-based security system for edge computing, *Proc. of the 13th International Conference on Availability, Reliability and Security*, pp.1-8, 2018.

[21] F. Jiang and H. W. Tseng, Trust model for wireless network security based on the edge computing, *Microsystem Technologies*, vol.27, no.4, pp.1627-1632, 2021.

[22] R. Islambouli, Z. Sweidan, A. Mourad et al., Towards trust-aware IoT Hashing offloading in mobile edge computing, *International Wireless Communications and Mobile Computing (IWCMC)*, pp.2216-2221, 2020.

[23] D. Wu, G. Shen, Z. Huang et al., A trust-aware task offloading framework in mobile edge computing, *IEEE Access*, vol.7, pp.150105-150119, 2019.

[24] X. Deng, J. Liu, L. Wang et al., A trust evaluation system based on reputation data in mobile edge computing network, *Peer-to-Peer Networking and Applications*, vol.13, no.5, pp.1744-1755, 2020.

[25] R. Latif, M. U. Ahmed, S. Tahir et al., A novel trust management model for edge computing, *Complex & Intelligent Systems*, vol.8, no.5, pp.3747-3763, 2022.

[26] X. Xu, Q. Wu, L. Qi et al., Trust-aware service offloading for video surveillance in edge computing enabled Internet of Vehicles, *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.3, pp.1787-1796, 2020.

[27] L. Zhang, Y. Zou, W. Wang, Z. Jin, Y. Su and H. Chen, Resource allocation and trust computing for blockchain-enabled edge computing system, *Computers and Security*, vol.105, 102249, 2021.

[28] J. Guo, A. Liu, K. Ota, M. Dong, X. Deng and N. N. Xiong, ITCN: An intelligent trust collaboration network system in IoT, *IEEE Transactions on Network Science and Engineering*, vol.9, pp.203-218, 2022.

[29] Y. Liu, J. Q. Zhu and J. Wang, Computation offloading optimization in mobile edge computing based on HIBSA, *Mobile Information Systems*, 2021.

[30] X. Wang, L. T. Yang, L. Ren et al., A tensor-based computing and optimization model for intelligent edge services, *IEEE Network*, vol.36, no.1, pp.40-44, 2022.

Author Biography

Jianhua Liu received the Ph.D. degree from Beihang University, Beijing, China, in 2013. He is currently an associate professor with the Institute of Electronic and Electrical Engineering, Civil Aviation Flight University of China, Guanghan, China. His research interest includes information security, Internet of Things, and edge computing. He is a member of Aviation Society of China. He won the Sichuan Flight Education Fund Award in 2023.

Xiaoni Shi received the B.S. degree in Communication Engineering from Tianjin University of Technology, China in 2020. She is currently pursuing the Master's degree with the Institute of Electronic and Electrical Engineering, Civil Aviation Flight University of China. She is conducting researches in the fields of mobile edge computing.

Jiajia Liu is an associate professor at the Institute of Electronic and Electrical Engineering, Civil Aviation Flight University of China. She received her B.S. and M.S. degrees in Communication and Information Systems from the Sichuan University in 2008 and 2011, respectively. Her current research interests include image processing, and cloud computing. She is member of China Aviation Society and Sichuan Electronics Society. She is also holding a fixed wing (Class IV) pilot license for beyond visual range vertical takeoff and landing, a civil aircraft maintenance personnel license, a Class II AV professional license for PA-44-180 (LO-360) aircraft, and concurrently serving as a police unmanned aerial vehicle flight instructor in Jining, Shandong Province.

Xiaoguang Tu is a lecturer at the Institute of Electronic and Electrical Engineering, Civil Aviation Flight University of China. He received his Ph.D. degree from the University of Electronic Science and Technology of China in 2020 and was a visiting scholar at the Learning and Vision Laboratory of the National University of Singapore from 2018 to 2020. His research interests include convex optimization, computer vision, deep learning and edge computing.