Goal-Driven Trusted Collaborator Selection and Task Offloading in Dynamic Collaborative Systems

Jiazhi Chen, *Graduate Student Member, IEEE*, Xianbin Wang, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Given the limited onboard resources and operational time constraints, dynamic collaboration among moving intelligent machines, such as unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) through task offloading has become essential for effective task completion. However, the growing offloading complexity and mismatch between task specifics and distributed resources inevitably lead to resource wastage and potential task failures. Furthermore, malicious collaborators may sneak into offloading processes, which undermines collaborative system reliability. To tackle these challenges collectively, a goal-driven trusted task offloading strategy is proposed, which efficiently matches diverse tasks to optimal distributed resources. Specifically, multidimensional goals of complex tasks are modeled as distinct task completion metrics, jointly termed Value of Service (VoS). Moreover, we define task-specific trust as a goal-achieving mechanism that enables the construction of a reliable collaborator group for a given task with diverse VoS. Based on the task-specific trust evaluation of all potential collaborators, the task offloading process is transformed into a trust-guided bipartite graph matching problem. To mitigate the matching complexity in large-scale collaborative systems, decomposed subtasks with similar goals are initially clustered into limited categories and subsequently arranged by priorities. Simulation results show the proposed strategy efficiently selects capable and reliable collaborators who complete tasks as expected in unreliable dynamic environments.**

***Index Terms*—Bipartite graph, dynamic collaborator selection, task offloading, trust, Value of Service (VoS).**

I. Introduction

MOVING intelligent machines, such as unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), have become essential in supporting various complex tasks, including traffic surveillance and disaster rescue, owing to their cost-effectiveness, high mobility, and versatile deployment capabilities [1]. However, the constraints on both onboard resources and operational time pose challenges for these machines to independently execute complex tasks, thereby necessitating dynamic collaboration through task offloading to resource-powerful machines for effective task completion [2]. Specifically, different from mobile edge computing servers and remote cloud, a fleet of moving intelligent machines concurrently possesses powerful computing and mobile capabilities, which could form a moving collaborator platform for efficiently completing delegated complex tasks [3], [4].

A. Motivation

Given the dramatically increased tasks and collaborators in large-scale collaborative systems, concurrently achieving efficient collaboration while guaranteeing task completion performance can be extremely challenging. This challenge arises primarily from the intricate requirements for capability and reliability during the selection of potential collaborators.

On the one hand, diverse completion goals associated with collaborative tasks lead to complex capability requirements when selecting potential collaborators. Traditional task offloading strategies typically aim to optimize a static and single completion goal, which results in resource wastage by blindly pursuing task-agnostic goals [5], [6], [7]. For instance, misallocating time-sensitive resources for time-tolerant tasks inevitably leads to significant wastage of time-related resources. Furthermore, reasonable tradeoffs among specific completion goals, such as completion time, energy consumption, and security, are essential to accurately reflect the holistic task goal for selecting reliable collaborators. More importantly, as most complex tasks consist of multiple dependent subtasks, their interdependencies necessitate corresponding connectivity among collaborators, introducing further complexity to the offloading process. While a collaborator with sufficient resources may complete a subtask successfully, inadequate contact duration with another collaborator assigned to a dependent subtask could still result in overall task failure.

On the other hand, concentrating solely on the capability of selected collaborators while neglecting their reliability still poses a significant risk to effective task completion, particularly in the presence of malicious collaborators. Given the inherent openness and dynamic nature of moving collaborative systems, the infiltration of malicious collaborators is inevitable, which initiates various security threats in task offloading and completion, encompassing both external and internal attacks [8]. While external attacks from unauthenticated collaborators could be mitigated through existing techniques like encryption, authentication, and digital signatures on remote security servers,

Received 11 September 2024; revised 30 October 2024; accepted 14 November 2024. Date of publication 18 November 2024; date of current version 26 March 2025. This work was supported in part by the New Frontiers in Research Fund of Government of Canada under Grant NFRFE-2022-00512; in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Program under Grant RGPIN-2024-05720; and in part by the Canada Research Chair Program. *(Corresponding author: Xianbin Wang.)*

Jiazhi Chen and Xianbin Wang are with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada (e-mail: jche629@uwo.ca; xianbin.wang@uwo.ca).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2024.3502006

these techniques generally fail to counter internal attacks by collaborators with legitimate and trustworthy identities [9], [10]. As task requestors typically lack control over the subsequent task execution process after offloading, these internal malicious collaborators, who always hide in the offloading process, could be selected as final collaborators, thereby compromising task completion performance.

Motivated by the aforementioned challenges, it is critical to design an efficient task offloading strategy that concurrently considers the dynamic capability and reliability requirements during collaborator selection. Given the variations in collaborator selection requirements induced by diverse tasks, designing distinct metrics that modeling multidimensional task completion goals is essential for optimal collaborator selection, collectively termed Value of Service (VoS). To this end, we define task-specific trust as a new goal-achieving mechanism, uniquely tailored to align collaborator selection with multidimensional task completion goals. As mentioned in our previous work [11], trust among moving machines is perceived as an aggregated confident indicator constituted of multiple factors related to whether a collaborator can complete tasks as expected. Here, we broaden this trust concept to task-specific trust by replacing completion expectation with a set of multidimensional goals to address the complex task offloading problem in dynamic collaborative systems.

Since complex tasks are often decomposed into multiple subtasks, we employ graph-based task representation to accurately characterize task structures and subtask interdependencies [12], [13], [14]. Likewise, the resource platform formed by moving collaborators is also modeled as a graph structure, wherein the resources of each collaborator are represented as a set of virtual machines (VMs) with the consideration of parallel resource occupancy [15], [16], [17]. Based on constructed task-resource graphs and defined task-specific trust, the offloading problem is converted to a trust-guided bipartite graph matching problem between subtasks and VMs. However, the exponentially growing complexity due to the collaborative system scale increment renders such graph matching ineffective or even infeasible. Therefore, designing an efficient task offloading strategy for large-scale collaborative systems is critical, and achievable by clustering subtasks into limited categories and arranging them by priorities.

B. Related Work

Task offloading in dynamic environments has been extensively explored in existing research to enhance task completion performance. Meanwhile, due to the openness and dynamic nature of collaborative systems inevitably initiating security threats, there is a growing reliance on trust mechanisms to detect and eliminate the participation of malicious collaborators.

1) Task Offloading in Dynamic Environments: Besides the strategies designed solely focusing on maximizing or minimizing a static and single task completion goal [5], [6], [7], an increasing number of studies have proposed new task offloading strategies in dynamic environments considering diverse task completion goals. Specifically, both task completion time and energy consumption are jointly optimized in [3] and [4] using utility functions, with game-theoretic and stable matching algorithms proposed for task-resource matching. However, these strategies mainly focus on static resource allocation and do not fully consider dynamic real-time changes. Moreover, in accommodating the demands of heavy computation of tasks, an energy-efficient collaborator group-assisted collaborative offloading strategy is proposed that splits tasks into multiple subtasks [18], but it lacks sufficient consideration for security in unreliable environments. To further analyze the dependency among subtasks, a graph-represented task scheduling problem is investigated in [12], using a low-complexity subgraph search and transmission power optimization, though this reliance on subgraph search can increase computational overhead in large-scale systems. Besides, recent reinforcement learning (RL)-driven task offloading strategies optimize collaborator selection through adaptive learning [19], [20], yet reliability concerns persist due to the risk of malicious collaborators. Although [21] explores using RL to mitigate these security risks, the complexity of RL-based strategy still poses a significant challenge in large-scale systems.

In a nutshell, given the high probability of infiltration by malicious collaborators in open and dynamic collaborative systems, security threats initiated by these malicious entities inevitably lead to unsatisfactory task completion performance and even task failure. Consequently, under such unreliable environments, the effectiveness of the existing offloading strategies mentioned above may be compromised when the reliability of collaborators is not considered.

2) Trust in Task Offloading: Inspired by the idea of leveraging trust to mitigate security threats in collaborative systems, several recent studies have sought to integrate trust into task offloading. For example, in [22], [23], and [24], trust is incorporated as an additional evaluation metric for collaborator selection alongside task completion time and energy consumption, aimed at detecting and mitigating the impact of malicious collaborators. Although these works can select reliable collaborators, their static evaluation of trust limits flexibility for tasks with evolving goals. Moreover, rather than isolating trust from other quality-of-service requirements, Kong et al. [25] and Rjoub et al. [26] proposed trust-based task offloading strategies that rely exclusively on evaluated trust values. Specifically, trust values are derived from metrics, such as average response time, average frequency ratio, and resource utilization of collaborators in [26]. Yet, these methods often require frequent updates of trust values based on collaborator performance, which escalates computational demands as the number of tasks and collaborators increases. Furthermore, by formulating the problem of dependent task offloading with bipartite graph matching, trust relationships can be incorporated into the graph as edge weights that need to be maximized, providing more effective solutions for matching tasks and resources [27], [28]. Nevertheless, the computational overhead associated with constructing and updating these graph models becomes substantial in large-scale systems.

In a nutshell, the trust mentioned in these existing works could not be adaptively adjusted according to the diverse goals of the tasks, which are primarily dedicated to static goals, such

as detecting and mitigating risks initiated by malicious collaborators. Moreover, existing trust-based offloading strategies often overlook the substantially increased computational overhead of trust evaluation in large-scale systems, stemming from frequent updates of trust values based on dynamic collaborator performance, which require extensive data processing as the number of collaborators and tasks grows.

C. Contributions and Organization

To efficiently match multiple tasks with diverse completion goals to capable and reliable resources under dynamic environments, we propose a goal-driven trusted task offloading strategy to accurately select optimal collaborator groups for effective task completion. The main contributions of this article are summarized as follows.

1) A multidimensional task completion goal model is proposed to achieve efficient task offloading tailored to diverse goals. By designing specific task completion metrics for diverse goals, collectively termed VoS, optimal collaborators are adaptively selected. Three task completion goals, including VoS of task completion time, VoS of energy consumption, and VoS of trust, are considered in this article. Specifically, the inherent openness of collaborative systems inevitably introduces unknown collaborators, making the VoS of trust essential to guarantee task completion reliability.
2) We define task-specific trust as a goal-achieving mechanism that assists the construction of reliable collaborator groups for complex tasks with varied VoS. By further modeling the task graph and resource graph, the process of dynamic task offloading to distributed resources is transformed from an adaptive mixed integer programming (MIP) problem into a trust-guided bipartite matching problem. In contrast to existing matching optimization problems with static task completion goals, the proposed problem could be adaptively adjusted according to diverse task completion goals.
3) To solve the trust-guided bipartite matching problem, we propose a goal-driven trusted task offloading strategy. Specifically, a task-specific trust evaluation method is proposed to accurately evaluate the trustworthiness of collaborators by adaptively aggregating task-specific historical information and third-party recommendations. To reduce the computational complexity of trust evaluation, an adaptive goal-driven subtask clustering (AGSC) algorithm is proposed that clusters subtasks into limited categories via similar goals. Besides, to guarantee intricate graph task structures, subtasks are arranged by priorities both within and across clusters through a proposed two-stage priority exploration algorithm.

The remainder of this article is organized as follows. Section II introduces the system model and problem formulation. Section III overviews the details of our proposed goal-driven trusted task offloading strategy. Simulation results are given in Section IV, followed by the conclusion in Section V.

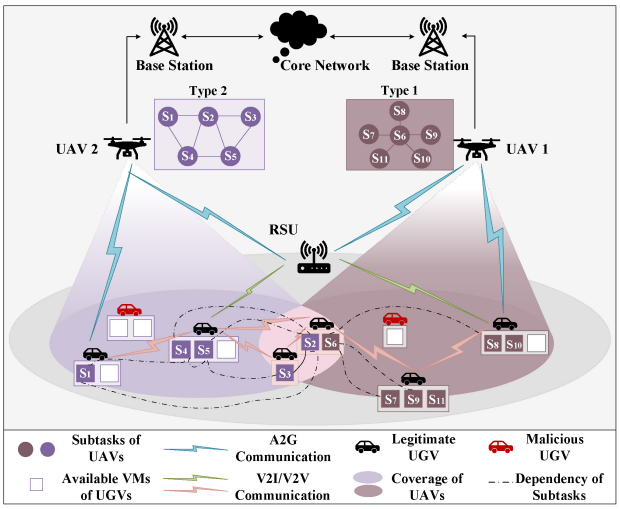

Fig. 1. Illustration of the UAV–UGV collaborative system, featuring two types of graph tasks to be offloaded from the UAV to UGVs. The UGV platform comprises a total of eight UGVs, with two designated as malicious.

II. System Model and Problem Formulation

In this section, we first represent complex UAV tasks and distributed UGV resources using the graph model, incorporating task-specific trust to efficiently associate task graphs and resource graphs. Assume that a central controller, such as a roadside unit (RSU), manages the UAV–UGV task offloading decisions. The UAV with a graph task transmits the required task completion goals to the central controller, which then evaluates and selects reliable collaborative UGVs. An illustrative example of the UAV–UGV collaborative system is presented in Fig. 1, considering two distinct types of UAV tasks, namely, type 1 with a star topology and type 2 with a bull topology. Besides, eight potential UGVs with available resources for collaboration are also depicted, with two random UGVs designated as malicious. The major system models are introduced in the following sections. The main notations used in this research are summarized in Table I.

A. UAV Task Model

The set of UAVs is denoted as $\mathcal{M} = \{1, \ldots, m, \ldots, M\}$, where each UAV $m \in \mathcal{M}$ owns a task that needs to be offloaded to UGVs for execution. Specifically, the task of UAV m is represented as a graph $\mathcal{G}_m = (\mathbf{S}_m, \Gamma_m, \mathbf{W}_m)$ [12], [13], [14], where $\mathbf{S}_m = \{s_{i,m} | i \in \{1, 2, \ldots, |\mathbf{S}_m|\}\}$ represents a set of dependent subtasks, and $|\mathbf{S}_m|$ indicates the total number of subtasks. The sets $\Gamma_m = \{\gamma_m^{i,i'} | i, i' \in \{1, 2, \ldots, |\mathbf{S}_m|\}, i \neq i'\}$ and $\mathbf{W}_m = \{w_m^{i,i'} | i, i' \in \{1, 2, \ldots, |\mathbf{S}_m|\}, i \neq i'\}$ denote the dependency and required contact duration between subtask $s_{i,m}$ and $s_{i',m}$, respectively. With multiple tasks from a set of UAVs, $\mathbf{S}_{\mathcal{M}} \triangleq \cup_{m \in \mathcal{M}} \mathbf{S}_m$ is the union set of all subtasks, which is formulated as $\mathbf{S}_{\mathcal{M}} = \{s_l | l = (i, m), i \in \{1, 2, .., |\mathbf{S}_m|\}, m \in \mathcal{M}\}$.

B. UGV Platform Model

The UGV platform consists of multiple UGVs denoted as the set $\mathcal{N} = \{1, \ldots, n, \ldots, N\}$, where each UGV n carries

TABLE I
MAJOR NOTATIONS AND DEFINITIONS

Notation	Definition
m	A UAV, $m \in \mathcal{M}$
n	A UGV, $n \in \mathcal{N}$
$s_{i,m}$	A subtask of a UAV graph task m, $s_{i,m} \in \mathbf{S}_m$
s_l	A subtask from a union set of all subtasks, $s_l \in \mathbf{S}_{\mathcal{M}}$
$\gamma_m^{i,i'}$	A dependency (edge) between i and i', $\gamma_m^{i,i'} \in \mathbf{\Gamma}_m$
$w_m^{i,i'}$	Required contact duration between i and i', $w_m^{i,i'} \in \mathbf{W}_m$
v_k	An available virtual machine, $v_k \in V_{\mathcal{N}}$
$\gamma_{n,n'}$	A one-hop communication link between n and n', $\gamma_{n,n'} \in \mathbf{\Gamma}$
$w_{n,n'}$	An exponential distribution parameter of contact duration between n and n', $w_{n,n'} \in \mathbf{W}$
$e_{i,k}$	A trust relationship between i and k, $e_{i,k} \in \mathbb{E}$
$w_{i,k}^g$	i's global trust value for k, $w_{i,k}^g \in \mathbb{W}$
$\hat{r}_m^p$	A completion goal (VoS) of a graph task, $\hat{r}_m^p \in \hat{\mathcal{R}}_m$
$\hat{r}_{i,m}^p$	A completion goal (VoS) of a subtask, $\hat{r}_{i,m}^p \in \hat{\mathcal{R}}_{i,m}$
$x_{i,k}$	Binary task offloading decision between i and k
$D_{i,m}$	Data size of a subtask
K	Total number of clusters
C_f	A cluster, $C_f \in C$, $f \in \{1, 2, ..., K\}$
c_f	A cluster centroid, $c_f \in \mathcal{C}$, $f \in \{1, 2, ..., K\}$
$\hat{r}_f^p$	A completion goal (VoS) of a cluster centroid, $\hat{r}_f^p \in \hat{\mathcal{R}}_f$
$w_{f,k}^c$	Capability trust value
$w_{f,k}^h$	Direct experiential trust value
$w_{f,k}^{in}$	Indirect trust value
$\mathcal{H}_{d,k}$	Historical collaboration records for direct experiential trust evaluation
$\mathcal{H}_{in,k}$	Historical collaboration records in indirect trust evaluation

multiple VMs providing available resources for completing subtasks [15], [16]. Similarly, we also model the UGV platform as an undirected graph $\mathcal{G} = (\mathcal{N}, \Gamma, \mathbf{W})$ [17], where $\mathcal{N}$ indicates the set of UGVs and each UGV $n \in \mathcal{N}$ owns a set of available VMs, formulated as $\mathbf{V}_n = \{v_{j,n} | j \in \{1, 2, \ldots, |\mathbf{V}_n|\}\}$. Moreover, $\Gamma = \{\gamma_{n,n'} | n, n' \in \mathcal{N}, n \neq n'\}$ and $\mathbf{W} = \{w_{n,n'} | n, n' \in \mathcal{N}, n \neq n'\}$ indicates the existence of the one-hop communication link between n and n' and corresponding edge weights served as exponential distribution parameters of contact duration between n and n'. With multiple VMs from different UGVs, $\mathbf{V}_{\mathcal{N}} \triangleq \cup_{n \in \mathcal{N}} \mathbf{V}_n$ is the union set of all available VMs, which can also be formulated as $\mathbf{V}_{\mathcal{N}} = \{v_k | k = (j, n), j \in \{1, 2, .., |\mathbf{V}_n|\}, n \in \mathcal{N}\}$.

C. Attack Model

With increased UGVs participating in collaborative systems, the potential infiltration of malicious UGVs is inevitable, leading to failed task offloading and even system collapse. Defending against internal attacks in collaborative systems is crucial due to the threat posed by malicious UGVs with legal and trustworthy identities inflicting greater system harm and detection challenges than external attacks. Specifically, two main internal attacks launched by malicious UGVs are considered in this study as follows.

1) *Blackhole Attack [29], [30]:* The malicious UGV falsely claims its capability to complete offloaded tasks, but upon receiving tasks and associated rewards from the UAV, discards the tasks without execution, thereby diminishing the success rate of task offloading. Such behavior of the malicious UGV is detected when task response time t_r is larger than the dwell time $T_{m,n}^d$, i.e., $t_r > T_{m,n}^d$, where $T_{m,n}^d = (d_{m,n}/\bar{v})$ ($d_{m,n}$ is the distance between the location of UGV n that start receiving the data at the beginning and the end point of the diameter of the circle in the forward direction of UGV n).
2) *On–Off Attack [31]:* The malicious UGV alternates between legitimate and malicious actions in an unpredictable pattern. It behaves normally for a period t_{legi}, during which it offloads tasks correctly, and randomly initiates a blackhole behavior during t_{mali}, where it drops all task information. The alternating behavior is represented as a random binary function, $f(t)$, where $f(t = t_{\text{legi}}) = 1$ during legitimate actions and $f(t = t_{\text{mali}}) = 0$ during malicious actions.
3) *Recommendation Attack [25]:* This attack also known as bad-mouthing or good-mouthing attacks, malicious recommenders send deceptive recommendations to task requestors, influencing the trust evaluation with misleading data. For example, let $w^g(A, B)$ denote the global trust rating of collaborator B as evaluated by A, while $w^g(C, B)$ is the deceptive rating sent by the malicious recommender C. The manipulated rating $w^g(C, B) \neq w_{\text{true}}^g(C, B)$ could mislead A, leading to incorrect trust decisions.

D. Trust Graph Model

The trust relationship among humans affects their decision to select the partner to cooperate with. In this article, we transfer the trust among humans to trust between UAVs and UGVs, where the trust is task-specific and gradually established with multiple collaborations. More specifically, the task-specific trust relationship between subtasks and VMs is harnessed for goal-driven task offloading. Such a trust relationship is modeled as a bipartite graph [27], [28], denoted as $\mathcal{G}_u = (\{\mathbf{S}_{\mathcal{M}}, \mathbf{V}_{\mathcal{N}}\}, \mathbf{E}_u, \mathbf{W}_u)$, where $\mathbf{S}_{\mathcal{M}}$ and $\mathbf{V}_{\mathcal{N}}$ signify the union set of all subtasks and all available VMs, respectively. Furthermore, $\mathbf{E}_u = \{e(s_l, v_k) | s_l \in \mathbf{S}_{\mathcal{M}}, v_k \in \mathbf{V}_{\mathcal{N}}\}$ indicates the edge (trust relationship) set of the trust graph. $\mathbf{W}_u = \{w^g(s_l, v_k) | s_l \in \mathbf{S}_{\mathcal{M}}, v_k \in \mathbf{V}_{\mathcal{N}}\}$ denotes the evaluated global trust value between subtask s_l and VM v_k. For a specific task of UAV m, let $\tilde{\mathcal{N}}$ represent the set of UGVs within the communication range of UAV m. Similarly, the set of trust relationship is denoted as $\mathbb{E} \subseteq \mathbf{E}_u$ and the corresponding global trust value set is represented as $\mathbb{W} \subseteq \mathbf{W}_u$, where $\mathbb{E} = \{e_{i,k} | s_{i,m} \in \mathbf{S}_m, k = (j, n), j \in \{1, 2, .., |\mathbf{V}_n|\}, n \in \tilde{\mathcal{N}}\}$ and $\mathbb{W} = \{w_{i,k}^g | s_{i,m} \in \mathbf{S}_m, k = (j, n), j \in \{1, 2, .., |\mathbf{V}_n|\}, n \in \tilde{\mathcal{N}}\}$, respectively.

E. Model of Multidimensional Task Completion Goals

The multidimensional completion goals of a UAV graph task $\mathcal{G}_m$ are represented as a set consisting of multiple features $\hat{\mathcal{R}}_m = \{\hat{r}_m^p | p \in \{1, 2, \ldots, |\hat{\mathcal{R}}_m|\}\}$. Due to each graph task $\mathcal{G}_m$ comprising a set of dependent subtasks $\mathbf{S}_m$ and each subtask $s_{i,m}$ also includes diverse completion goals $\hat{\mathcal{R}}_{i,m} = \{\hat{r}_{i,m}^p | s_{i,m} \in$

$\mathbf{S}_m, p \in \{1, 2, \ldots, |\hat{\mathcal{R}}_m|\}$. Inspired by [32] and [33], a new task completion metric, i.e., VoS, is harnessed to model diverse completion goals. We assume the VoS of each subtask $s_{i,m}$ and corresponding graph task $\mathcal{G}_m$ are known by the RSU, which is normalized as a value within the range [0,1]. By leveraging VoS, diverse and potentially conflicting goals across different tasks can be modeled separately, such as time and security requirements, enabling flexible adaptation to multitask collaborative systems. Overall, three expected subtask VoS are considered as they represent essential dimensions of task completion performance, including the VoS of subtask completion time $\hat{r}^1_{i,m}$, the VoS of subtask energy consumption $\hat{r}^2_{i,m}$, and the VoS of subtask trust $\hat{r}^3_{i,m}$. The corresponding expected VoS of the overall graph task $\mathcal{G}_m$ follows: $\hat{r}^1_m = \min \sum_{i=1}^{|\mathbf{S}_m|} \hat{r}^1_{i,m}$, $\hat{r}^2_m = \min \sum_{i=1}^{|\mathbf{S}_m|} \hat{r}^2_{i,m}$, and $\hat{r}^3_m = (1/|\mathbf{S}_m|) \sum_{i=1}^{|\mathbf{S}_m|} \hat{r}^3_{i,m}$. Details of the actual VoS calculation are given as follows.

1) VoS of Task Completion Time: The completion time calculation of a subtask $T_{i,m}$ needs to consider task structure. When offloading two connected subtasks $s_{i,m}$ and $s_{i',m}$ to different UGVs n and n', $T_{i,m}$ encompasses transmission time, data exchange time $\Delta\tau_{n,n'}$, and computation time $t^C_{i,k}$. Otherwise, $T_{i,m}$ only comprises transmission time and computation time. A piece-wise function indicates $T_{i,m}$ as

$$T_{i,m} = \begin{cases} \sum_{j=1}^{|\mathbf{V}_n|} \sum_{n=1}^{|\tilde{\mathcal{N}}|} \left(\frac{x_{i,k} \times D_{i,m}}{\phi_{m,n}} + \Delta\tau_{n,n'} + t^C_{i,k} \right) \\ \text{if } \gamma^{i,i'}_m = 1, x_{i,k} \times x_{i',k'} = 1 \\ \forall v_k \in n, v_{k'} \in n', n \neq n' \\ \sum_{j=1}^{|\mathbf{V}_n|} \sum_{n=1}^{|\tilde{\mathcal{N}}|} \left(\frac{x_{i,k} \times D_{i,m}}{\phi_{m,n}} + t^C_{i,k} \right) \\ \text{otherwise} \end{cases} \tag{1}$$

where $D_{i,m}$ is the data size of subtask $s_{i,m}$, $x_{i,k}$ indicates the binary task offloading decision and $\phi_{m,n}$ indicates the data transmission rate between UAV m and the UGV n that the offloaded VM v_k belong to, given by

$$\phi_{m,n} = B \log_2\left(1 + q_{m,n}\varphi_{m,n}\right) \tag{2}$$

where B denotes the channel bandwidth, $q_{m,n}$ is the transmission power between UAV m and UGV n, and $\varphi_{m,n}$ represents the varying channel quality. Similar to [34] and [35], we model $\varphi_{m,n}$ as a continuous random variable, including factors, such as fading, path loss, and noise, which obeys a uniform distribution in the interval $[\varepsilon_1, \varepsilon_2]$, denoted by $\varphi_{m,n} \sim \mathrm{U}(\varepsilon_1, \varepsilon_2)$. Thus, $q_{m,n}\varphi_{m,n}$ represents the received signal-to-noise ratio (SNR) according to Shannon's theorem, where a higher value of $q_{m,n}\varphi_{m,n}$ enables a larger transmission rate.

In terms of the data exchange time $\Delta\tau_{n,n'}$ between UGV n and n', an opportunistic V2V contact model is harnessed, where $\Delta\tau_{n,n'}$ obeys exponential distribution [36], [37]. Therefore, the probability of the data exchange time between UGV n and n' should be larger than $\Delta t = |(D_{i,m}/\phi_{m,n}) - (D_{i',m}/\phi_{m,n'})| + w^{i,i'}_m$, denoted as $\Pr(\Delta\tau_{n,n'} \geq \Delta t | w_{n,n'}) = e^{-\Delta t \times w_{n,n'}}$.

Without loss of generality, the actual VoS of subtask completion time $r_{i,1}$ is modeled by harnessing the following two functions [33], [38]: 1) a sigmoidal function for time-sensitive subtasks and 2) a logarithmic function for time-tolerant subtasks, formulated as

$$r^1_{i,m} = \begin{cases} \frac{1}{1+e^{\alpha_1(T_{i,m}-\alpha_2)}}, & \text{if } \varpi_{i,m} = 1 \\ 1 - \alpha_3 \log\left(1 + \alpha_4 T_{i,m}\right), & \text{if } \varpi_{i,m} = 0 \end{cases} \tag{3}$$

where the binary variable $\varpi_{i,m}$ is used to describe the type of subtask $s_{i,m}$ toward time-related characteristic, $\varpi_{i,m} = 1$ with $0.5 < \hat{r}^1_{i,m} \leq 1$ indicates time-sensitive subtasks and $\varpi_{i,m} = 0$ with $0 < \hat{r}^1_{i,m} \leq 0.5$ indicates time-tolerant subtasks, respectively.

Based on the subtask completion time in (1), the overall completion time T_m of a graph UAV task $\mathcal{G}_m$ is calculated with the slowest subtask completion time, given by

$$T_m = \max \sum_{i=1}^{|\mathbf{S}_m|} T_{i,m} \tag{4}$$

thus the actual VoS of task completion time is denoted as $r^1_m = \min \sum_{i=1}^{|\mathbf{S}_m|} r^1_{i,m}$.

2) VoS of Energy Consumption: UAVs incur extra offloading overhead induced by transmitting data with UGVs and hovering in the sky to wait for task results. We ignore the energy waste from the transmission as it is typically much smaller than that from hovering, which is even smaller by two orders of magnitude [39], [40]. Therefore, the UAV energy consumption by solely offloading a subtask $s_{i,m}$ is denoted as

$$E_{i,m} = q^f_m T_{i,m} \tag{5}$$

where q^f_m indicates the propulsion power of the UAV m. A thrust-based propulsion energy consumption model is utilized to evaluate the propulsion power q^f_m [40], [41].

Due to the limited computing capacities and energy resources of UAVs, increased UAV energy consumption inescapably impairs task completion performance. Thus, the actual VoS of UAV energy consumption for offloading a subtask $s_{i,m}$ is formulated as a sigmoidal function within the range (0, 1], given by

$$r^2_{i,m} = \frac{1}{1 + e^{\beta_1(E_{i,m} - \beta_2)}}. \tag{6}$$

Accordingly, the overall UAV energy consumption for offloading the whole graph task $\mathcal{G}_m$ is formulated as

$$E_m = q^f_m \max \sum\nolimits_{i=1}^{|\mathbf{S}_m|} T_{i,m} \tag{7}$$

where the actual VoS of UAV energy consumption for offloading the whole graph task is denoted as $r^2_m = \min \sum_{i=1}^{|\mathbf{S}_m|} r^2_{i,m}$.

3) VoS of Trust: Traditionally, trust from one entity to another is dedicated to its legitimate identity authentication as well as secure communication without data leakage [42]. However, the task-specific trust concept in this article is not limited to identity security or communication security but further expands to collaborator group selection security, ensuring the task completion performance is within the soft tolerance of task requestors. Such broadening of the trust concept is crucial due to the high likelihood of internal malicious collaborator participation in dynamic collaborative systems, where task requestors typically cannot predict whether such collaborators could process the tasks as expected [11], [24]. Furthermore, malicious behaviors initiated by certain collaborators may be

temporary, attributed to uncontrollable external factors such as poor channel quality. In such cases, these collaborators may still deliver satisfactory services in the future. Consequently, we propose a new task completion metric, namely, VoS of trust, where the VoS of trust for a subtask $s_{i,m}$ indicates the probability of the selected VM v_k completing the subtask according to various goals, ranging from 0 to 1, given by

$$r^3_{i,m}=\begin{cases} w^g_{i,k}+w^g_{i,k}\lambda_r\frac{|\hat{\theta}_{i,m}-\theta_{i,m}|}{\hat{\theta}_{i,m}}, & \theta_{i,m}\geq\hat{\theta}_{i,m}, \\ w^g_{i,k}-w^g_{i,k}\lambda_p\frac{|\hat{\theta}_{i,m}-\theta_{i,m}|}{\hat{\theta}_{i,m}}, & \theta_{i,m}<\hat{\theta}_{i,m} \end{cases} \tag{8}$$

where $w^g_{i,k}$ indicates the global trust value of a VM v_k from subtask $s_{i,m}$ perspective, reflecting the probability of achieving satisfactory completion of subtask $s_{i,m}$. In general, when no historical task offloading has occurred between UAV m and VM v_k, the initial trust value $w^g_{i,k}$ of VM v_k is set to 0.5. Subsequently, the trust value $w^g_{i,k}$ is dynamically updated upon collaboration. The update process of trust value depends on the comparison between the actual subtask utility $\theta_{i,m}$ and the expected subtask utility $\hat{\theta}_{i,m}$, where we assume $\theta_{i,m}=(r^1_{i,m}+r^2_{i,m})/2$ and $\hat{\theta}_{i,m}=(\hat{r}^1_{i,m}+\hat{r}^2_{i,m})/2$. λ_r and λ_p indicate the reward and penalty weights, respectively. Based on the calculated VoS of subtask trust in (8), the actual VoS of trust for the whole graph task $\mathcal{G}_m$ is denoted as $r^3_m=(1/|\mathbf{S}_m|)\sum_{i=1}^{|\mathbf{S}_m|}r^3_{i,m}$.

F. Problem Formulation

With the involvement of the system models above, the overall objective of this work is summarized as minimizing the discrepancy between expected VoS and actual VoS by selecting a reliable collaborator group for a given task with diverse goals, where the actual VoS should always be "just-above" the expected VoS. Given $\mathbf{x}=[x_{i,k}]_{1\leq i\leq|\mathbf{S}_m|,1\leq k\leq|\mathbf{V}_{\mathcal{N}'}|,1\leq m\leq M}$ that denote the matrix of binary variable $x_{i,k}$, we capture the tradeoff between VoS of task completion time, VoS of energy consumption, and VoS of trust by function $\mathcal{R}(\mathbf{x})$, defined as

$$\mathcal{R}(\mathbf{x})=\frac{1}{M}\sum_{m=1}^{M}(\omega_1|r^1_m-\hat{r}^1_m|+\omega_2|r^2_m-\hat{r}^2_m| +\omega_3|r^3_m-\hat{r}^3_m|) \tag{9}$$

where a weighted sum of the discrepancy between expected VoS and actual VoS is calculated. Considering a large amount of uncertainty in the UAV–UGV collaborative system including but not limited to malicious UGV participation, varying air-to-ground (A2G) channel quality, V2V contact duration, and UAV energy budget, the proposed optimization problem for goal-driven task offloading is formulated as

$$\boldsymbol{\mathcal{P}}:\arg\min_{\mathbf{x}}\mathcal{R}(\mathbf{x}) \tag{10}$$

s.t.

$$r^p_m\geq\hat{r}^p_m,\ r^p_{i,m}\geq\hat{r}^p_{i,m}\ \forall\hat{r}^p_m\in\hat{\mathcal{R}}_m,\hat{r}^p_{i,m}\in\hat{\mathcal{R}}_{i,m} \tag{C1}$$

$$\sum_{j=1}^{|\mathbf{V}_n|}\sum_{n=1}^{|\tilde{\mathcal{N}}|}\frac{x_{i,k}\times D_{i,m}}{\phi_{m,n}}\leq T^d_{m,n} \quad \forall s_{i,m}\in\mathbf{S}_m, m\in\mathcal{M}, n\in\tilde{\mathcal{N}} \tag{C2}$$

$$e^{-\Delta t\times w_{n,n'}}\geq\mu,\ \gamma^{i,i'}_m=1, x_{i,k}\times x_{i',k'}=1, n\neq n' \quad \forall v_k\in n, v_{k'}\in n', m\in\mathcal{M}, n\in\mathcal{N} \tag{C3}$$

$$\sum_{j=1}^{|\mathbf{V}_n|}\sum_{n=1}^{N}\sum_{i=1}^{|\mathbf{S}_m|}x_{i,k}\times q_{m,n}\leq Q_m\ \forall m\in\mathcal{M} \tag{C4}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{|\mathbf{S}_m|}x_{i,k}\leq|\mathbf{V}_{\mathcal{N}}|\ \forall n\in\mathcal{N} \tag{C5}$$

where C1 guarantees the actual VoS of every goal should always be above the expected VoS. Constraint C2 only allows UAVs to transmit tasks to VMs belong the UGVs that are within the communication range of UAVs due to the communication link between the UAVs and UGVs must remain connected until data transmission is completed. Constraint C3 ensures that when offloading two connected subtasks $s_{i,m}$ and $s_{i',m}$ of a graph task $\mathcal{G}_m$ to different UGVs n and n', the probability of the data exchange time between n and n' that is larger than Δt should be greater than a threshold μ within range (0,1]. Constraint C4 controls the UAV energy budget associated with offloaded tasks. Constraint C5 limits the offloaded subtasks to exceed the available number of VMs.

The proposed optimization problem in (10) indicates an adaptive MIP problem with binary offloading indicator $x_{i,k}$. Solving the problem is challenging in guaranteeing the graph task structure and task completion performance through appropriate subtask-VM matching, as stated in C1 and C3. Traditionally, exhaustive-search-based methods are utilized to solve such an MIP problem [43], [44], inevitably bringing prohibitive nondeterministic polynomial (NP) complexity. Moreover, the difficulty of solving the problem also arises from the unreliable task completion performance stemming from the extensive involvement of malicious collaborators. To achieve efficient subtask-VM matching with reliable task completion performance, we employ graph theory that integrates the task-specific trust relationship between subtask and VM [27], [28], i.e., convert (10) to a trust-guided bipartite graph matching problem solved in polynomial time. The solving strategy for this trust-guided bipartite graph matching problem is described below.

III. Goal-Driven Trusted Task Offloading

The overall process of our proposed goal-driven trusted task offloading strategy is depicted in Fig. 2, encompassing four main steps outlined in the following sections. By analyzing multidimensional task completion goals and evaluating specific trust value between subtask and VM, this strategy effectively reduces the complexities associated with matching multiple UAV tasks featuring distinct goals to UGVs offering diverse capabilities. It is extensible to accommodate tasks with divergent or even conflicting goals through goal-driven clustering and priority exploration.

A. Adaptive Goal-Driven Subtask Clustering

Given the exponential growth in the number of tasks and potential collaborators in dynamic collaborative systems, the

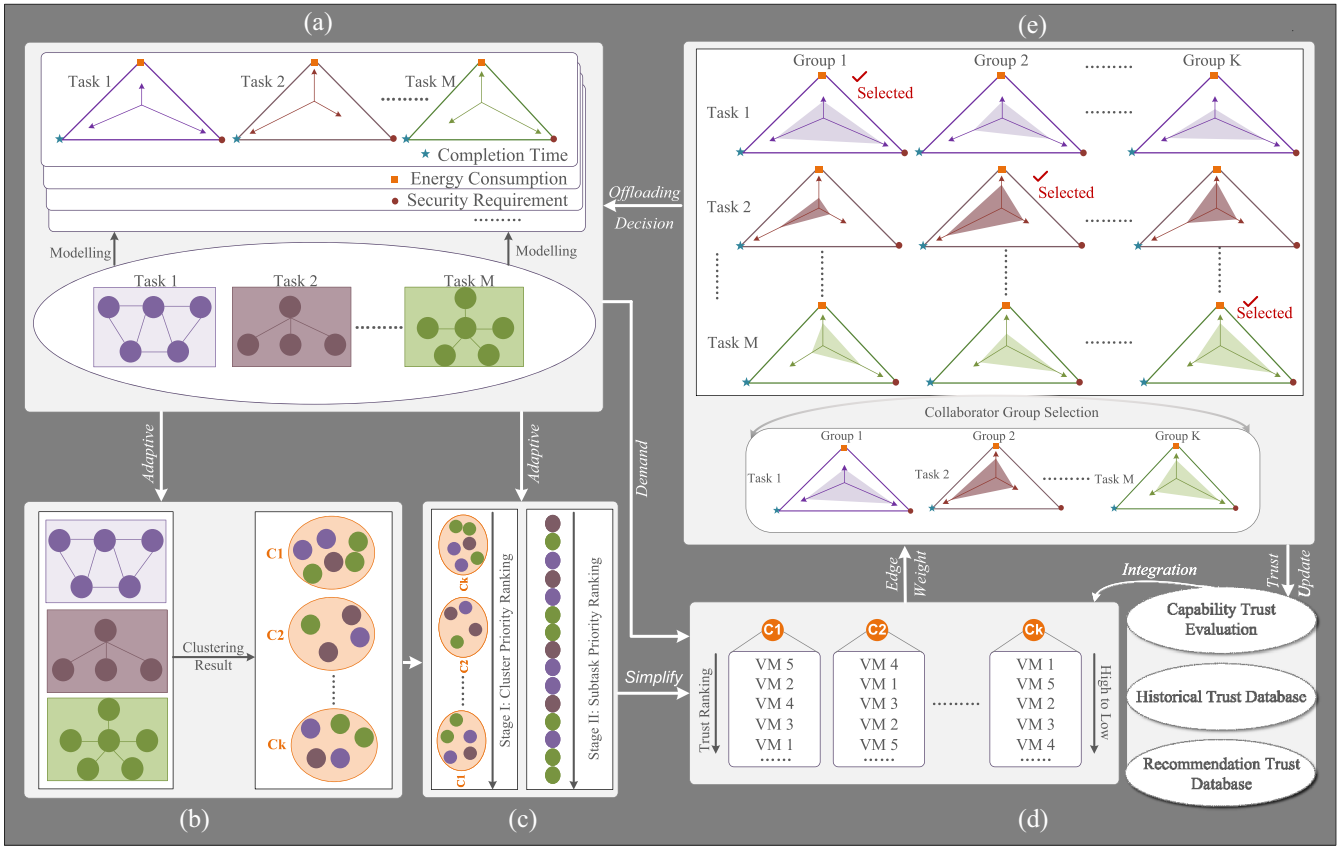

Fig. 2. Detailed steps of the proposed goal-driven trusted task offloading strategy. Based on (a) modeling multidimensional task completion goals, subtask-VM offloading decisions are then obtained through (b) AGSC, (c) two-stage subtask priority exploration, (d) task-specific trust evaluation, and (e) trust-guided bipartite graph matching-based collaborator group selection. Note that the trust value is adaptively updated and stored as historical information for the next offloading process.

analysis of diverse task completion goals and the evaluation of collaborators become increasingly complex. To reduce such complexity, an AGSC algorithm is first proposed, which clusters concurrent subtasks in the UAV network according to similar goals, which is illustrated in Algorithm 1.

The main difference between the proposed AGSC algorithm and the traditional K-means clustering algorithm lies in the adaptive selection of the K clusters according to the subtask completion goals. The proposed AGSC algorithm comprises two main steps, namely, optimal cluster selection and adaptive subtask similarity calculation. Aiming to decide the optimal clusters $K_{\min} \leq K \leq K_{\max}$ of the union set of subtask $\mathbf{S}_{\mathcal{M}}$ according to their completion goals $\cup_{m\in\mathcal{M}}\hat{\mathcal{R}}_{i,m}$, we employ the average silhouette method that find the optimal K by maximizing the average silhouette coefficient [45], [46]. The silhouette coefficient is calculated by comparing the similarity of a subtask to its own cluster with its neighboring clusters, denoted as $[(b(\hat{\mathcal{R}}_{i,m}) - a(\hat{\mathcal{R}}_{i,m}))/(\max(a(\hat{\mathcal{R}}_{i,m}), b(\hat{\mathcal{R}}_{i,m})))]$. More detailed definition of the silhouette coefficient can be found in [45]. Then, the average silhouette coefficient is obtained by calculating the mean of silhouette coefficients over all subtasks, given by

$$\vartheta_K = \frac{\sum_1^{|\mathbf{S}_{\mathcal{M}}|} \frac{b(\hat{\mathcal{R}}_{i,m}) - a(\hat{\mathcal{R}}_{i,m})}{\max(a(\hat{\mathcal{R}}_{i,m}), b(\hat{\mathcal{R}}_{i,m}))}}{|\mathbf{S}_{\mathcal{M}}|}. \tag{11}$$

Subsequently, the set of clusters $C = \{C_f | f \in \{1, 2, \ldots, K\}\}$ that yields the largest value of the average silhouette coefficient in set Λ is obtained as the optimal cluster selection. To accurately assign subtasks with similar completion goals to the same cluster C_f, $K = K_{\min}$ is selected as the initial number of clusters and a set of centroids $\mathcal{C} = \{c_f | f \in \{1, 2, \ldots, K\}\}$ is randomly selected from the subtasks union set. Then calculate the similarity through Euclidean distance evaluation between selected centroids and other subtasks according to subtask VoS, and assign each subtask to the cluster with the closest centroid, denoted as

$$\phi(s_l) = \underset{f}{\operatorname{argmin}} ||\hat{\mathcal{R}}_{i,m} - \hat{\mathcal{R}}_f||_2^2 \tag{12}$$

where $\hat{\mathcal{R}}_f = \{\hat{r}_f^p | f \in \{1, 2, \ldots, K\}, p \in \{1, 2, \ldots, |\hat{\mathcal{R}}_m|\}\}$ indicates the completion goals (VoS) of each centroid c_f, $\phi(s_l)$ represents the clustering label of a subtask s_l.

The cluster centroids are continuously updated by taking the mean of all subtask completion goals assigned to each cluster, given by

$$c_f = \frac{1}{|C_f|} \sum_{s_l \in C_f} \hat{\mathcal{R}}_{i,m} \tag{13}$$

where c_f stops updating until convergence, i.e., equal to the previous centroid c_{pre}.

Algorithm 1: AGSC

Input: Union set of all subtasks $\mathbf{S}_{\mathcal{M}}$ and corresponding completion goals $\cup_{m\in\mathcal{M}}\hat{\mathcal{R}}_{i,m}$, minimum and maximum number of clusters $K_{\min}$ and $K_{\max}$

Output: Set of optimal clusters C

Initialization:
$\Lambda \leftarrow \emptyset, K = K_{\min}$
while $K \leq K_{\max}$ **do**
  // Random selection of centroids
  foreach $c_f \in \mathcal{C}$ **do**
    $c_f \leftarrow s_l \in \mathbf{S}_{\mathcal{M}}$
  Repeat
    foreach $s_l \in \mathbf{S}_{\mathcal{M}}$ **do**
      foreach $c_f \in \mathcal{C}$ **do**
        Assign subtasks to clusters with minimum similarity distance based on (12)
    foreach $c_f \in \mathcal{C}$ **do**
      $c_{pre} \leftarrow c_f$
      Update cluster centroids c_f based on (13)
  Until $c_{pre} == c_f$
  Calculate average silhouette value ϑ_K based on (11)
  $\Lambda \leftarrow \vartheta_K, K = K + 1$
return Optimal clusters with max Λ

B. Two-Stage Subtask Priority Exploration

To avoid unnecessary costs induced by coordinating conflicts among subtasks that select the same collaborative VM, the process of exploring priority among subtasks becomes essential. Based on the clustered result of subtasks concerning their different completion goals, the priority degree of each cluster should be explored as the first stage, which is concretized below.

Definition 1 (Priority Degree of Subtask Cluster $\mathcal{D}(C_f)$): The priority degree of subtask cluster $\mathcal{D}(C_f)$ is defined as the weighted sum of the feature vector of corresponding cluster centroids, namely, the weighted sum of all subtask completion goals (VoS), given by

$$\mathcal{D}(C_f) = \sum_{p=1}^{|\hat{\mathcal{R}}_m|} \omega_p \hat{r}_f^p$$

where $\hat{r}_f^p$ and ω_p denote one of the subtask completion goals and corresponding weight. Besides, the set of subtask cluster priority degrees is denoted as $\boldsymbol{\mathcal{D}(C)}$, which is sorted in descending order.

In the next stage, the priority order of subtasks in each cluster is first determined to preserve the structure of each graph task. Starting from the cluster C_f with the highest degree $\mathcal{D}(C_f)$, the order of subtasks within the cluster is analyzed. We assume a total of J UAV graph task in C_f is known by RSU and define an empty set Z_j as the priority set of subtasks belonging to the same graph task $\mathcal{G}_j$ in the cluster C_f, where the subtask s_l with the largest weighted sum of subtask completion goals $\sum_{p=1}^{|\hat{\mathcal{R}}_m|} \omega_p \hat{r}_l^p$ is added as the first component in Z_j. Then, the other subtasks in the cluster will be added one by one into the set Z_j based on two situations, depending on whether they belong to the same graph task as the first component. The corresponding priority policy of other subtasks in the same cluster C_f is concretized below.

Definition 2 (Priority Policy of Subtasks): The priority policy of subtasks in same cluster C_f is divided into two situations below.

1) Belong to the same graph task $\mathcal{G}_j$, the priority degree of the subtask s_l is defined as the total number of edges connected to itself in graph task $\mathcal{G}_m$, denoted as $\mathcal{D}_1(s_l)$.
2) Not belong to the same graph task $\mathcal{G}_j$, the priority degree of a subtask is defined as the weighted sum of subtask completion goals (VoS), denoted as $\mathcal{D}_2(s_l)$.

Based on the above definitions, the priority sequence $\mathcal{Z}$ of all subtasks can be explored, where the procedure of our proposed two-stage subtask priority exploration algorithm is summarized in Algorithm 2.

C. Task-Specific Trust Evaluation

A task-specific trust evaluation method is proposed to assess the trustworthiness of various VMs, namely, determining their capability and reliability to complete the different subtasks. Specifically, three trust factors of potential VM are accurately evaluated, including capability trust, direct experiential trust, and indirect trust, and then adaptively aggregated to obtain the global trust value of VM, which served as the corresponding edge weight in the trust bipartite graph. A large number of subtasks are clustered into limited categories based on the proposed AGSC algorithm, thus the overall trust evaluation process of potential VMs is dedicated to the subtask VoS of each cluster centroid and subtasks belonging to the same cluster share the same collaborative VM ranking.

1) Capability Trust Evaluation: Capability trust value $w_{f,k}^c$ indicates the estimation of a VM v_k's current capability to complete a target subtask $c_f \in \mathcal{C}$, which is calculated by

$$w_{f,k}^c = \begin{cases} 1 - \sum_{p=1}^{|\hat{\mathcal{R}}_m|} \omega_p (r_f^p - \hat{r}_f^p), & r_f^p \geq \hat{r}_f^p \\ 0, & r_f^p < \hat{r}_f^p \end{cases} \tag{14}$$

where $\hat{r}_f^p$ and r_f^p denote an expected and actual VoS that the evaluated VM v_k could provide for subtask cluster centroid c_f. The difference $r_f^p - \hat{r}_f^p$ illustrates the task-specific trust concept, assigning higher trust value to collaborators suited for the task over those with higher capabilities. A larger capability trust value indicates the VM v_k has higher capabilities to complete the subtask.

2) Direct Experiential Trust Evaluation: Direct experiential trust evaluation $w_{f,k}^h$ is a prediction of VM v_k's current collaboration behavior based on historical statistics within the same cluster C_f. This is applicable only if there are previous collaboration experiences between UAVs in cluster C_f and VM v_k; otherwise, the evaluation is skipped.

Let $\mathcal{H}_{d,k} = \{(\mathcal{R}_d(t), \hat{\mathcal{R}}_d(t)) | d \in \{1, 2, .., D\}, t \in \{1, 2, .., T\}, k = (j, n), j \in \{1, 2, .., |\mathbf{V}_n|\}, n \in \mathcal{N}\}$ denotes the historical collaboration records, which is represented in Fig. 3. Here, $\mathcal{R}_d(t)$ is the actual performance of VM v_k in completing

Algorithm 2: Two-Stage Subtask Priority Exploration

Input: Union set of all subtasks $\mathbf{S}_{\mathcal{M}} = \{s_l | l \in \{1, 2, ..., |\mathbf{S}_{\mathcal{M}}|\}\}$ and corresponding completion goals $\cup_{m\in\mathcal{M}} \hat{\mathcal{R}}_{i,m}$

Output: Priority sequence $\mathcal{Z}$ of all subtasks

Initialization:

$\boldsymbol{\mathcal{D}(C)} \leftarrow \emptyset, Z_j \leftarrow \emptyset, \mathcal{Z} \leftarrow \emptyset$, calculate $\mathcal{D}_1(s_l)$ and $\mathcal{D}_2(s_l)$ for all $s_l \in \mathbf{S}_{\mathcal{M}}$

// Stage I: Subtask cluster priority exploration

foreach $C_k \in C$ **do**

  $\boldsymbol{\mathcal{D}(C)} \leftarrow$ Calculate $\mathcal{D}(C_k)$

  Sort $\boldsymbol{\mathcal{D}(C)}$ in a descending order

  Sort C by following the order of $\boldsymbol{\mathcal{D}(C)}$

// Stage II: Intercluster subtask priority exploration

for $f = 1 : K$ **do**

  for $j = 1 : J$ **do**

    $Z_j \leftarrow$ Add the subtask $s_l \in C_f$ with the largest value of $\mathcal{D}_2(s_l)$

    foreach $s_{l'} \in C_f \cap s_{l'} \notin Z_j$ **do**

      if $s_l, s_{l'} \in \mathcal{G}_j, l \neq l'$ **then**

        $Z_j \leftarrow$ Add the subtask $s_{l'} \in C_f$ with the largest value of $\mathcal{D}_1(s_l)$

        Sort Z_j based on $\mathcal{D}_1(s_l)$ in descending order

      else

        $\mathbb{Z} \leftarrow$ Add $s_{l'} \in C_f$

    $C_f \leftarrow$ Update cluster $C_f \cap \mathbb{Z}$

  $\mathcal{Z} \leftarrow Z_1$

  for $j = 2 : J$ **do**

    for $i = 1 : |Z_j|$ **do**

      if $i == 1$ **then**

        $\mathcal{Z} \leftarrow$ Add the subtask $s_i \in Z_j$

        Sort $\mathcal{Z}$ in descending order

      else

        $\mathbb{V}' \leftarrow [\mathcal{Z}.index(i-1) : |Z_j|]$, $\mathbb{V} \leftarrow [1 : \mathcal{Z}.index(i-1)]$

        $\mathbb{V}' \leftarrow$ Add the subtask $s_i \in Z_j$

        Sort $\mathbb{V}'$ in descending order

    $\mathcal{Z} \leftarrow \mathbb{V} \cup \mathbb{V}'$

a subtask offloaded by UAV d at the tth collaboration, while $\hat{\mathcal{R}}_d(t)$ refers to the corresponding expected subtask completion goals. These records capture both the actual and expected outcomes for subtasks performed by VM v_k for UAV d during prior collaborations.

In this context, $d \in D$ refers to a UAV in cluster C_f that has previously collaborated with VM v_k, and $t \in T$ indicates the specific collaboration instance. Based on these historical collaboration records $\mathcal{H}_{d,k}$, the direct experiential trust evaluation is modeled using the beta distribution [47], [48], which is formulated as

UAVs in C_f - D	Collaboration Instance - T		
	Collaboration$_{t=1}$	Collaboration$_{t=2}$	Collaboration$_{t=T}$
UAV$_{d=1}$	$(\mathcal{R}_1(1), \widehat{\mathcal{R}}_1(1))$	$(\mathcal{R}_1(2), \widehat{\mathcal{R}}_1(2))$	$(\mathcal{R}_1(T), \widehat{\mathcal{R}}_1(T))$
UAV$_{d=2}$	$(\mathcal{R}_2(1), \widehat{\mathcal{R}}_2(1))$	$(\mathcal{R}_2(2), \widehat{\mathcal{R}}_2(2))$	$(\mathcal{R}_2(T), \widehat{\mathcal{R}}_2(T))$
...	...	...	...
UAV$_{d=D}$	$(\mathcal{R}_D(1), \widehat{\mathcal{R}}_D(1))$	$(\mathcal{R}_D(2), \widehat{\mathcal{R}}_D(2))$	$(\boldsymbol{\mathcal{R}_D(T), \widehat{\mathcal{R}}_D(T)})$

Fig. 3. Illustration of historical collaboration records $\mathcal{H}_{d,k}$ for direct experiential trust evaluation, also adaptable to historical collaboration records of indirect trust evaluation within cluster $C_{f'}$ by substituting $d \in D$ and $t \in T$ with $d' \in D'$ and $t' \in T'$.

$$w^h_{f,k} = \Upsilon^h \frac{\sum \mathcal{H}^+_{d,k} + 1}{\sum \mathcal{H}^+_{d,k} + \sum \mathcal{H}^-_{d,k} + 2} \tag{15}$$

where Υ^h indicates the task similarity of direct experiential trust evaluation, that is, the squared Euclidean distance between the current and all previous subtask completion goals ($\hat{\mathcal{R}}_d(t)$ and $\hat{\mathcal{R}}_f$) of direct experiential trust evaluation, given by

$$\Upsilon^h = \frac{1}{D \times T} \sum_{d=1}^{D} \sum_{t=1}^{T} ||\hat{\mathcal{R}}_d(t) - \hat{\mathcal{R}}_f||_2^2 \tag{16}$$

and $\sum \mathcal{H}^+_{d,k}$ indicates the total number of positive records when the actual performance $\mathcal{R}_d(t)$ meets or exceeds the expected performance $\hat{\mathcal{R}}_d(t)$, and vice versa for $\sum \mathcal{H}^-_{d,k}$.

3) Indirect Trust Evaluation: Indirect trust value $w^{in}_{f,k}$ refers to the recommendations from third parties, i.e., the UAV with subtasks in other clusters $C_{f'} \in C, f \neq f'$, regarding the trustworthiness of a VM v_k. When a UAV in cluster C_f lacks sufficient information for evaluating direct experiential trust of VM v_k, recommendations from other clusters $C_{f'}$ compensate for the lack of direct collaboration experiences.

Similarly, let $\mathcal{H}_{\text{in},k} = \{(\mathcal{R}^{f'}_{d'}(t'), \hat{\mathcal{R}}^{f'}_{d'}(t')) | f' \in \{1, 2, .., K-1\}, d' \in \{1, 2, .., D'\}, t' \in \{1, 2, .., T'\}, k = (j, n), j \in \{1, 2, .., |\mathbf{V}_n|\}, n \in \mathcal{N}\}$ denote the historical collaboration records from recommenders in other clusters $C_{f'}$. In this context, $\mathcal{R}^{f'}_{d'}(t')$ and $\hat{\mathcal{R}}^{f'}_{d'}(t')$ represent the actual subtask completion performances of v_k and expected subtask completion goals, both provided by UAV d' from cluster $C_{f'}$ during the t'th collaboration. These historical records from third-party recommendations serve as the basis for calculating the indirect trust value of VM v_k, which is calculated as

$$w^{in}_{f,k} = \Upsilon^{in} \frac{\sum \mathcal{H}^+_{\text{in},k} + 1}{\sum \mathcal{H}^+_{\text{in},k} + \sum \mathcal{H}^-_{\text{in},k} + 2} \tag{17}$$

where Υ^{in} indicates the task similarity of indirect trust evaluation, ensures that recommendations from UAVs handling tasks similar to those in cluster C_f are weighted more heavily, reducing the impact of biased recommendations, given by

$$\Upsilon^{in} = \frac{1}{(K-1) \times D' \times T'} \sum_{f'=1}^{K-1} \sum_{d'=1}^{D'} \sum_{t'=1}^{T'} ||\hat{\mathcal{R}}^{f'}_{d'}(t') - \hat{\mathcal{R}}_f||_2^2 \tag{18}$$

where $\mathcal{R}^{f'}_{d'}(t') \geq \hat{\mathcal{R}}^{f'}_{d'}(t')$, positive collaboration records are updated, denoted as $\sum \mathcal{H}^+_{\text{in},k}$. Conversely, negative records

are updated in $\sum \mathcal{H}^{-}_{\text{in},k}$. The beta distribution in the indirect trust calculation balances these records with smoothing factors (1 and 2) to prevent an over-reliance on extreme recommendations.

4) Adaptive Trust Factor Aggregation: Depending on whether direct historical collaboration and recommendations exist, four different situations are considered to calculate the global trust value of v_k, given by

1) c_1: $\mathcal{H}_{d,k} = \emptyset$ & $\mathcal{H}_{\text{in},k} = \emptyset$,
2) c_2: $\mathcal{H}_{d,k} \neq \emptyset$ & $\mathcal{H}_{\text{in},k} = \emptyset$,
3) c_3: $\mathcal{H}_{d,k} = \emptyset$ & $\mathcal{H}_{\text{in},k} \neq \emptyset$,
4) c_4: $\mathcal{H}_{d,k} \neq \emptyset$ & $\mathcal{H}_{\text{in},k} \neq \emptyset$.

Based on different situations, the current global trust value $w^g_{f,k}$ is formulated as

$$w^g_{f,k} = \begin{cases} w^c_{f,k}, & c_1 \\ \varsigma_1 w^c_{f,k} + (1-\varsigma_1) w^h_{f,k}, & c_2 \\ \varsigma_2 w^c_{f,k} + (1-\varsigma_2) w^{in}_{f,k}, & c_3 \\ \varsigma_3 w^c_{f,k} + \varsigma_4 w^h_{f,k} + \varsigma_5 w^{in}_{f,k}, & c_4 \end{cases} \tag{19}$$

where ς_1, ς_2, ς_3, ς_4, and ς_5 denote different weights for diverse trust evaluation factors.

D. Trust-Guided Bipartite Graph Matching

With the trust values of each VM evaluated for each subtask, the strategy proceeds to form collaboration VM groups to complete the overall UAV graph task. Potential VMs for completing the corresponding subtask are sorted based on their trust values in descending order, ensuring that the ones with maximum trust value are selected for collaboration, which is considered as a maximum bipartite graph matching problem. However, both the existence of the one-hop A2G communication between subtasks and VMs and the dependency among different subtasks could severely affect the matching results, thus the matched VM should meet both A2G communication and graph task structure constraints. In a nutshell, the edge weight in the trust bipartite graph, i.e., global trust value, needs to be further adjusted by following the definition below.

Definition 3 (Edge Weight of Trust Bipartite Graph): The edge weight (trust value) $\mathbf{W}_u$ of the trust bipartite graph $\mathcal{G}_u$ is adjusted following two conditions below.

1) For a collaborative subtask s_l requested by UAV m, if VM v_k located out of the communication range of UAV m, the global trust value of v_k to complete s_l is adjusted to zero.
2) If the offloading between subtask s_l and VM v_k fails to meet all edge and weight constraints of the subtask s_l, the global trust value of v_k to complete s_l is adjusted to zero.

Based on the priority sequence of subtasks $\mathcal{Z}$, the procedure of the optimal collaborator group selection through bipartite graph matching is summarized in Algorithm 3, where the VM with the maximum trust value is selected for task offloading.

After selecting the optimal collaborator group for each UAV graph task, the strategy proceeds to offload the tasks to these chosen groups for execution. Upon task completion and the return of results, the trust values are updated following the satisfaction level of the completed tasks. The trustworthiness of the collaborative UGVs can either increase with a reward weight λ_r or decrease with a penalty weight λ_p, as indicated in (8), depending on their alignment with the task goals, thus further adjusting the future task offloading strategy based on updating historical information.

Algorithm 3: Trust-Guided Maximum Bipartite Graph Matching for Collaboration Group Selection

Input: Subtask priority sequence $\mathcal{Z}$, $\mathbf{S}_\mathcal{M}$, $\mathbf{V}_\mathcal{N}$, and $\mathbf{W}_u$ in trust bipartite graph $\mathcal{G}_u$
Output: Optimal offloading decisions $\mathbf{X}_u$
Initialization: $\mathbf{X}_u \leftarrow []$, edge weight matrix $[\mathcal{W}_u]_{\mathcal{M}\times\mathcal{N}}$
for $z = 1 : |\mathcal{Z}|$ **do**
  Get the edge weight list $[\mathcal{W}_u]_{z\times\mathcal{N}}$ of the zth component in $\mathcal{Z}$
  Assign the zth component to the VM v_k with the maximum edge weight in $[\mathcal{W}_u]_{z\times\mathcal{N}}$
  $\mathcal{X} \leftarrow$ corresponding subtask-VM pair
  Adjust the edge weight of the selected VM v_k in $[\mathcal{W}_u]_{\mathcal{M}\times\mathcal{N}}$ to zero
  $\mathbf{X}_u \leftarrow \mathbf{X}_u \cup \mathcal{X}$

E. Computational Complexity Analysis

The proposed goal-driven trusted task offloading strategy achieves lower complexity with a simplified matching process compared to traditional bipartite graph matching using the Hungarian algorithm [27], which typically requires $O(|\mathbf{S}_\mathcal{M}|^3)$ due to solving a complete bipartite graph matching problem. Specifically, clustering subtasks based on their completion goals reduces the problem size, resulting in a clustering complexity of $O(|\mathbf{S}_\mathcal{M}|KI)$, where I is the number of iterations till convergence. The task-specific trust evaluation and trust-guided bipartite graph matching process operating on these clusters K both have a reduced complexity of $O(K\mathbf{V}_\mathcal{N})$, where $K \ll |\mathbf{S}_\mathcal{M}|$. After combined with the subtask priority exploration complexity of $O(|\mathbf{S}_\mathcal{M}| \log |\mathbf{S}_\mathcal{M}|)$, the dominant complexity of the proposed scheduling strategy becomes $O(|\mathbf{S}_\mathcal{M}| \log |\mathbf{S}_\mathcal{M}| + O(K\mathbf{V}_\mathcal{N}))$, significantly lower than the cubic complexity of traditional matching strategy.

IV. Performance Evaluation

In this section, a series of simulations are conducted to evaluate the performance of our proposed goal-driven trusted task-offloading strategy. The simulation validates the effectiveness of the proposed offloading strategy in achieving guaranteed task completion performance under different settings, including various graph tasks, UGV platform structures, proportions of malicious UGVs, as well as task completion goals.

A. Simulation Setup

The simulation considers an area of 1000 m (length) × 1000 m (width) × 100 m (height) consisting of multiple UAVs and UGVs that are randomly distributed. Two types of graph tasks requested by UAVs are considered as shown

TABLE II
MAJOR SIMULATION PARAMETERS

Parameters	Value
Data size of subtask $D(s_{i,m})$	[500, 600] Kb [12]
A2G channel bandwidth B	[6, 8] MHz [34]
Varying channel quality $\varphi_{m,n}$	U(100, 400) [35]
Transmission power $q_{m,n}$	550 mW [35]
Propulsion power $q_{f,m}$	100 W [40]
Maximum energy of UAV Q_m	1×10^5 Joule [41]
Average velocity of UGVs $\bar{v}$	12 m/s [4]
Computation time of VM $\theta(s_{i,m})$	[100, 200] ms [12]

in Fig. 1, where the required connect duration $w_m^{i,i'}$ between dependent subtasks is randomly selected within the range [0.1, 0.3]. The A2G communication with varying channel quality $\varphi_{m,n}$, which follows uniform distribution U(100, 400) [35]. In terms of the settings of the UGV platform, the subtask computation time of each VM is randomly selected within [100, 200] ms [12]. The edge weight $\gamma_{n,n'}$ settings of the UGV platform belong to [0.05, 0.06] in small problem sizes and [0.01, 0.02] in large problem sizes that reflect different V2V connection duration. Besides, some weighted parameter is defined as $\omega_1{=}\omega_2{=}\omega_3{=}1/3$, and $\mu \in [0.9, 1)$. To assess the attack resistance and the performance of trust evaluation, malicious collaborators engage in attack behaviors outlined in Section II-C. The proportion of malicious UGVs varies between 20% and 50% of the total UGVs, utilizing a combination of On–Off and recommendation attacks as hybrid threats. Specifically, during the On–Off attack, malicious UGVs alternate between legitimate and malicious actions in cycles of 20 task offloading rounds. Critical parameters are summarized in Table II.

B. Evaluation Metrics and Comparison Methods

We analyze the performance of the proposed goal-driven trusted task offloading strategy using the following metrics: 1) *running time of the task offloading strategy*; 2) *value of the proposed objective function* $\mathcal{R}(\mathbf{x})$; 3) *attack resistance rate*, which is defined as the ratio of unselected malicious VMs to the total number of malicious VMs; and 4) *actual VoS*, which is the weighted sum of r_m^1, r_m^2, and r_m^3.

To make the experimental results more convincing, four different baselines are compared with the proposed task offloading strategy.

1) *Time Heuristic Strategy (THS) [49]:* The priority policy of subtasks follows the descending ranking of the VoS of task completion time. The collaborator selection for each subtask relies solely on their task completion time, which is selected with the shortest completion time.
2) *VoS Maximization Oriented Strategy (VMOS) [32]:* The priority policy of subtasks follows the descending ranking of the weighted sum VoS concerning task completion time, energy consumption, and security. Each subtask is offloaded to the collaborator with the maximum VoS.

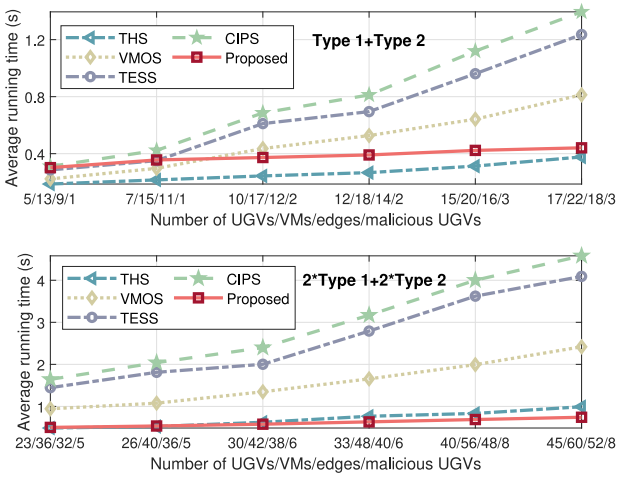

Fig. 4. Average running time comparison of various task offloading strategies, where the top subplot indicates the small problem sizes while the bottom subplot indicates the large problem sizes.

3) *Trust-Based Exhaustive Search Strategy (TESS) [11]:* The priority policy of subtasks is the same as that of the proposed strategy switches between edge degree and weighted sum of subtask VoS. The trust evaluation method only considers time-sensitive tasks and evaluates all VMs without subtask clustering consideration, where the VM with the maximum trust value is selected as the collaborator.
4) *Clustering Ignored Proposed Strategy (CIPS):* This strategy is designed to compare our proposed strategy without considering subtask clustering, where the proposed trust evaluation method exhaustively evaluates all VMs and the VM with the minimum difference between trust value and subtask VoS is selected as the collaborator.

C. Running Time Performance

In Fig. 4, the running time trends for various task offloading strategies are comprehensively illustrated. The top subplot delves into small problem sizes, while the bottom subplot explores large problem sizes featuring an increased number of UGVs, VMs, and V2V connections (edges), which are selected based on the number and topology of graph tasks [12], [17], focusing on star and bull topologies, ensuring that each subtask can select a VM and maintaining a 20% proportion of malicious UGVs across both small and large problem sizes [25], [29]. Overall, the running time of all task-offloading strategies increases with a larger problem size. Notably, the running time for both TESS and CIPS sharply increases due to the complex priority ranking of subtasks and exhaustive evaluation of potential collaborators, rendering them less suitable for large-scale scenarios. Besides, it can be observed that the running time of both THS and VMOS is pretty low in small problem sizes due to simpler subtask priority ranking but still exhibits a gradual rise with larger problem sizes. In contrast, by leveraging intelligent subtask clustering to simplify the evaluation process of potential

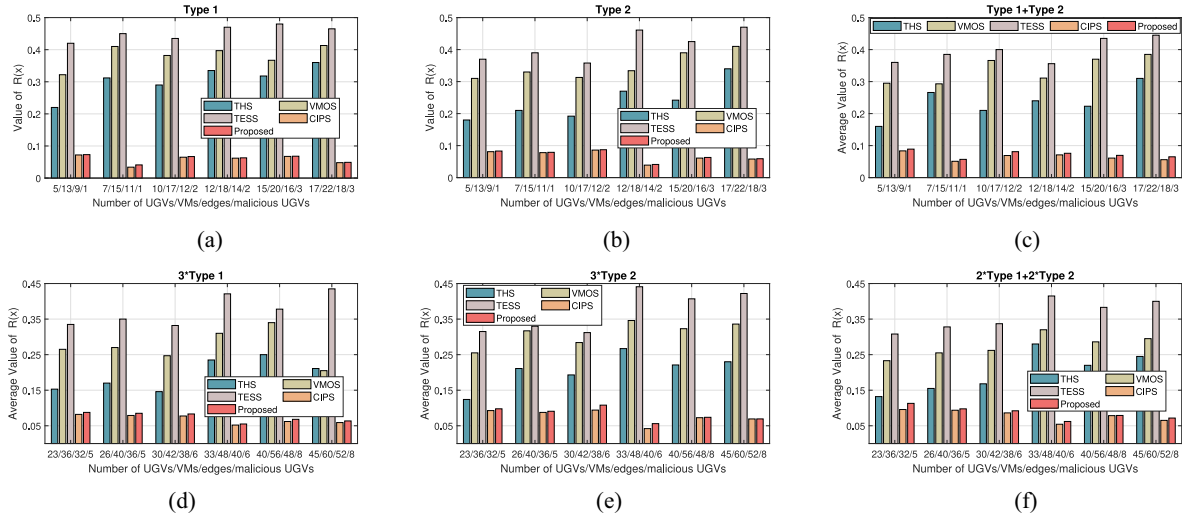

Fig. 5. Performance comparison of the value of $\mathcal{R}(\mathbf{x})$ in various problem sizes, where the subfigures (a)–(c) represent small problem sizes, while (d)–(f) correspond to large problem sizes. Besides, the title of each subfigure denotes the utilized graph task type(s).

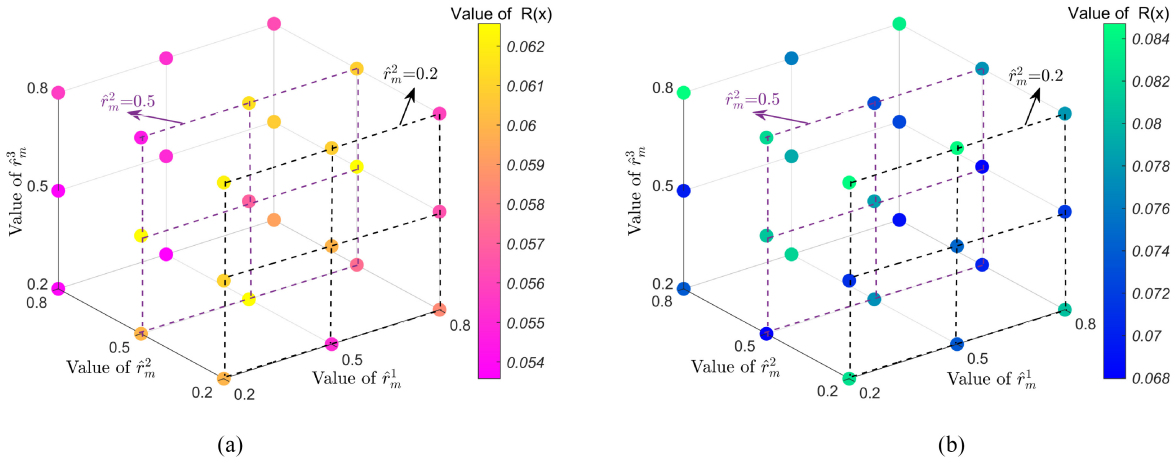

Fig. 6. Performance comparison of the value of $\mathcal{R}(\mathbf{x})$ with randomly selected task completion goals ($\hat{r}_m^1$, $\hat{r}_m^2$, $\hat{r}_m^3$) from [0.2, 0.5, 0.8], totalling 27 combinations. (a) Small problem size with Type 1+Type 2, number of UGVs/VMs/edges/malicious UGVs: 7/15/11/1. (b) Large problem size with 2*Type 1+2*Type 2, number of UGVs/VMs/edges/malicious UGVs: 40/56/48/8.

collaborators, our proposed strategy maintains a stable running time trend, showcasing its efficiency in handling larger scale networks.

D. Value of $\mathcal{R}(\mathbf{x})$ in Different Problem Sizes and Task Completion Goals

Fig. 5 depicts performance comparisons between the baselines and our proposed strategy concerning the value of $\mathcal{R}(\mathbf{x})$ across varying problem sizes. Here, we set uniform task completion goals: $\hat{r}_m^1 = 0.5$, $\hat{r}_m^2 = 0.5$, and $\hat{r}_m^3 = 0.5$. The average $\mathcal{R}(\mathbf{x})$ value is obtained across $\mathcal{R}(\mathbf{x})$ value of multiple tasks when more than one UAV task is requested. Our proposed strategy consistently outperforms THS, VMOS, and TESS, achieving significantly lower values of $\mathcal{R}(\mathbf{x})$ in both small [Fig. 5(a)–(c)] and large problem sizes [Fig. 5(d)–(f)]. This superior performance is attributed to the fact that these three baselines are best-effort-based offloading strategies, which select collaborators without considering task completion goals. Notably, both CIPS and our proposed strategy achieve relatively low values of $\mathcal{R}(\mathbf{x})$. While CIPS, a modified version of our proposed strategy, can achieve a slightly lower $\mathcal{R}(\mathbf{x})$ compared to our proposed strategy by exhaustively evaluating collaborators, its running time becomes prohibitively large for larger problem sizes.

To demonstrate the robustness of our proposed strategy, UAV graph tasks with diverse task completion goals are considered for task offloading, where the values of $\hat{r}_m^1$, $\hat{r}_m^2$, and $\hat{r}_m^3$ are randomly selected from the range [0.2, 0.5, 0.8], resulting in a total of 27 combinations. Fig. 6 illustrates performance comparisons of the value of $\mathcal{R}(\mathbf{x})$ with different task completion goals across varying problem sizes. In each subfigure, a color bar is incorporated to match the colors of the circles in the 3-D plot, representing the corresponding value of $\mathcal{R}(\mathbf{x})$. As shown in Fig. 6(a), the value of $\mathcal{R}(\mathbf{x})$ varies from 0.054 to 0.062 under different task completion goals for small problem sizes. In contrast, Fig. 6(b) demonstrates a slight increase in the value of $\mathcal{R}(\mathbf{x})$ within the range of

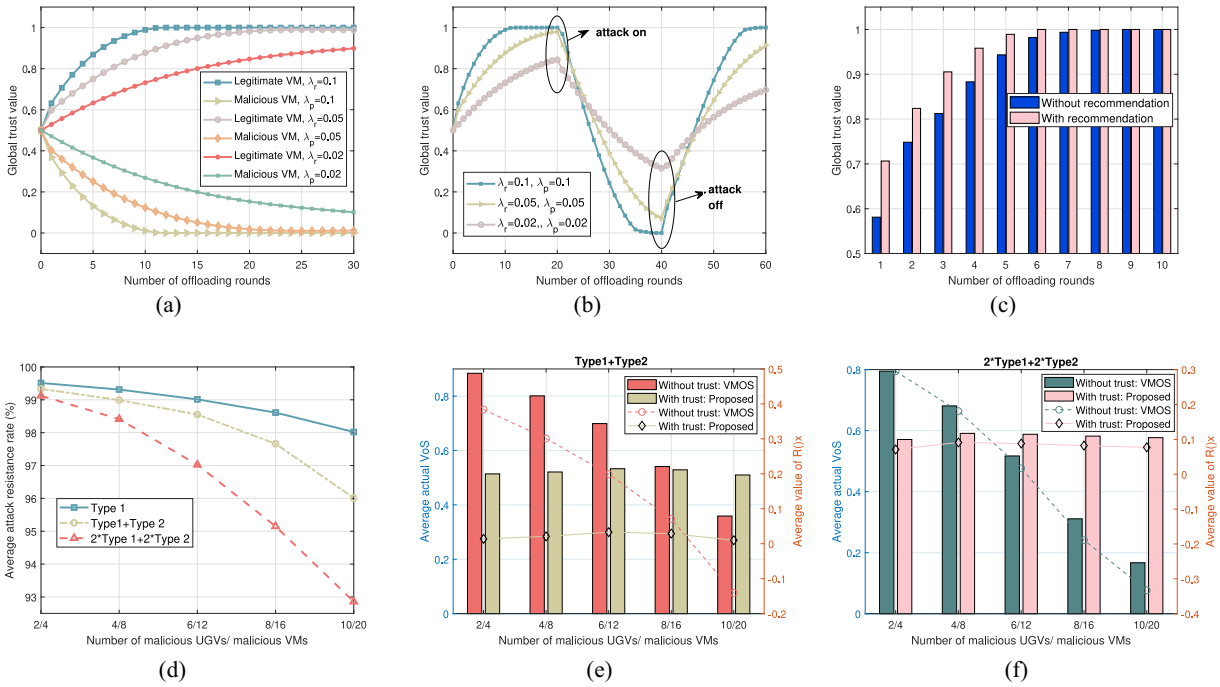

Fig. 7. Attack resistance and trust evaluation performance. (a) Global trust value of legitimate VM and malicious VM under different reward weight λ_r and penalty weight λ_p. (b) Update trend of global trust value under on–off attack, where attack initiates on 20th rounds and stops on 40th rounds. (c) Effect of recommendation from third parties (indirect trust) on global trust value update. (d) Average attack resistance rate under different numbers of malicious UGV and VMs. (e) For Type 1+Type 2, the average actual VoS and the average value of $\mathcal{R}(\mathbf{x})$ under various numbers of malicious UGVs/VMs. (f) For 2*Type 1+2*Type 2, the average actual VoS and the average value of $\mathcal{R}(\mathbf{x})$ under various numbers of malicious UGVs/VMs.

0.068 to 0.084, indicating that our proposed strategy always selects collaborators accurately that meet the specified task completion goals.

E. Attack Resistance and Trust Evaluation Performance

In larger network scenarios, the risk of collaborating with malicious UGVs increases, potentially compromising task completion performance. To address this, the proposed trust evaluation method incorporates a reward and penalty mechanism, as shown in Fig. 7(a), where legitimate collaborative VMs experience a steady rise in global trust value as their task completion performance aligns with goals. In contrast, the global trust value of malicious VMs decreases over time. Trust updates are influenced by the reward weight λ_r and penalty weight λ_p, with higher weights leading to faster adjustments. Fig. 7(b) further validates the on–off attack resistance of the proposed trust evaluation method, demonstrating that when a UGV initiates an attack in the 20th offloading round and ceases by the 40th, the method effectively adapts to the transitions between legitimate and malicious behaviors. Furthermore, as shown in Fig. 7(c), integrating third-party recommendations into the trust evaluation process significantly accelerates the update of the global trust value compared to relying exclusively on direct information, such as direct experiential trust and capability trust. For a legitimate VM, the global trust value swiftly reaches 1 by the 6th round with the aid of recommendations, while it only reaches the same level by the 9th round without them.

To evaluate the attack resistance performance of the proposed task offloading strategy, varying numbers of malicious UGVs and VMs are introduced into the system for task offloading, while the total number of UGVs, VMs, and edges are uniformly set to 30, 42, and 38. As illustrated in Fig. 7(d), the average attack resistance rate is obtained over 100 offloading rounds, revealing a decrease in resistance as the number of malicious UGVs and VMs increases. Specifically, with a higher number of requested UAV graph tasks, the average attack resistance rate tends to decrease. Nevertheless, even under the scenario of 10 malicious UGVs and 20 malicious VMs for 2*Type 1 + 2*Type 2 (a total of 22 subtasks), the average attack resistance rate remains at 93%, demonstrating the robust attack resistance performance of the proposed task offloading strategy.

With varying numbers of malicious UGVs and VMs, it is essential to compare the actual task completion performance to further validate the attack resistance capability of the proposed task offloading strategy. As depicted in Fig. 7(e) and (f), where uniform task completion goals ($\hat{r}_m^1 = 0.5$, $\hat{r}_m^2 = 0.5$, $\hat{r}_m^3 = 0.5$) are set, the average actual VoS on the left y-axis label represents the task completion performance, calculated by averaging the weighted sum of r_m^1, r_m^2, and r_m^3 of each task. We compare two task offloading strategies: one with trust, our proposed strategy, and one without trust, represented by VMOS, which does not satisfy constraint (C1). The average actual VoS of the without-trust strategy VMOS sharply decreases due to its lack of an integrated attack resistance mechanism. In contrast, the actual VoS of the proposed strategy remains around 0.5 in Fig. 7(e) and 0.57 in Fig. 7(f), aligning with the task completion goals. The average value of $\mathcal{R}(\mathbf{x})$ on the right y-axis label of Fig. 7(e) and (f) is approximately 0.01 and 0.07, respectively, further validating

that the proposed strategy can achieve a perfect alignment with task completion goals even under large-scale attacks.

V. Conclusion

In this article, we address the complex task offloading problem in dynamic collaborative systems, focusing on tasks with diverse completion goals and resources with varied capabilities and reliability. We transform the offloading problem into a trust-guided bipartite matching problem by introducing task-specific trust as a goal-achieving mechanism that optimizes collaborator selection based on specific task goals. The proposed task-specific trust offers valuable insights for optimizing any collaborative operations with specific goals, laying the groundwork for future research in intelligent task orchestration within collaborative systems. The dramatically increased matching complexity in large-scale collaborative systems is mitigated via the proposed adaptive subtask clustering and two-stage subtask priority exploration algorithm. Numerical results demonstrate that the proposed goal-driven trusted task offloading strategy achieves more efficient task completion by selecting "just-suitable" collaborator groups compared with other existing strategies. In the future, we will integrate artificial intelligence into trust-based task-offloading strategies to address the key limitations in potential edge scenarios, such as heterogeneous task requirements, fluctuating resource availability, and diverse security attacks.

References

[1] M. Dai, T. H. Luan, Z. Su, N. Zhang, Q. Xu, and R. Li, "Joint channel allocation and data delivery for UAV-assisted cooperative transportation communications in post-disaster networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16676–16689, Sep. 2022.

[2] X. Qin et al., "Timeliness-oriented asynchronous task offloading in UAV-edge-computing systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 1, pp. 900–912, Jan./Feb. 2024.

[3] G. Sun et al., "Joint task offloading and resource allocation in aerial-terrestrial UAV networks with edge and fog computing for post-disaster rescue," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8582–8600, Sep. 2024.

[4] Y. Wang et al., "Task offloading for post-disaster rescue in unmanned aerial vehicles networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1525–1539, Aug. 2022.

[5] Z. Hu, F. Zeng, Z. Xiao, B. Fu, H. Jiang, and H. Chen, "Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1630–1645, Apr.–Jun. 2021.

[6] S. Liu et al., "Dependent task scheduling and offloading for minimizing deadline violation ratio in mobile edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 538–554, Feb. 2023.

[7] L. X. Nguyen, Y. K. Tun, T. N. Dang, Y. M. Park, Z. Han, and C. S. Hong, "Dependency tasks offloading and communication resource allocation in collaborative UAV networks: A metaheuristic approach," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 9062–9076, May 2023.

[8] Z. Liu et al., "Lightweight trustworthy message exchange in unmanned aerial vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2144–2157, Feb. 2023.

[9] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13784–13795, Nov. 2020.

[10] Y. Song, T. Feng, C. Yang, X. Mi, S. Jiang, and M. Guizani, "IS2N: Intent-driven security software-defined network with blockchain," *IEEE Netw.*, vol. 38, no. 3, pp. 118–127, May 2024.

[11] J. Chen, X. Wang, and X. S. Shen, "RTE: Rapid and reliable trust evaluation for collaborator selection and time-sensitive task handling in Internet of Vehicles," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 12278–12291, Apr. 2024.

[12] M. Liwang, Z. Gao, S. Hosseinalipour, Y. Su, X. Wang, and H. Dai, "Graph-represented computation-intensive task scheduling over air-ground integrated vehicular networks," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3397–3411, Sep./Oct. 2023.

[13] M. Shafiee and J. Ghaderi, "Scheduling coflows with dependency graph," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, pp. 450–463, Feb. 2022.

[14] J. Li, B. Gu, Z. Qin, Z. Lin, and Y. Han, "DQN-based computation-intensive graph task offloading for Internet of Vehicles," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2022, pp. 1797–1802.

[15] G. Gao, M. Xiao, J. Wu, H. Huang, S. Wang, and G. Chen, "Auction-based VM allocation for deadline-sensitive tasks in distributed edge cloud," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1702–1716, Nov./Dec. 2021.

[16] D. Alboaneen, H. Tianfield, Y. Zhang, and B. Pranggono, "A metaheuristic method for joint task scheduling and virtual machine placement in cloud data centers," *Future Gener. Comput. Syst.*, vol. 115, pp. 201–212, Feb. 2021.

[17] Z. Liu, Y. Zhao, S. Hosseinalipour, Z. Gao, L. Huang, and H. Dai, "TDRA: A truthful dynamic reverse auction for DAG task scheduling over vehicular clouds," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 4337–4351, Mar. 2024.

[18] H. Zeng et al., "USV fleet-assisted collaborative computation offloading for smart maritime services: An energy-efficient design," *IEEE Trans. Veh. Technol.*, vol. 73, no. 10, pp. 14718–14733, Oct. 2024.

[19] S. Chen, L. Rui, Z. Gao, W. Li, and X. Qiu, "Cache-assisted collaborative task offloading and resource allocation strategy: A metare-inforcement learning approach," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19823–19842, Oct. 2022.

[20] J. Huang, J. Wan, B. Lv, Q. Ye, and Y. Chen, "Joint computation offloading and resource allocation for edge-cloud collaboration in Internet of Vehicles via deep reinforcement learning," *IEEE Syst. J.*, vol. 17, no. 2, pp. 2500–2511, Jun. 2023.

[21] Y. Zhou, G. Cheng, Y. Zhao, Z. Chen, and S. Jiang, "Toward proactive and efficient DDoS mitigation in IIoT systems: A moving target defense approach," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2734–2744, Apr. 2022.

[22] H. Long, C. Xu, G. Zheng, and Y. Sheng, "Socially-aware energy-efficient task partial offloading in MEC networks with D2D collaboration," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 3, pp. 1889–1902, Sep. 2022.

[23] P. Dass and S. Misra, "DeTTO: Dependency-aware trustworthy task offloading in vehicular IoT," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24369–24378, Dec. 2022.

[24] J. Zhang, B. Gong, M. Waqas, S. Tu, and Z. Han, "A hybrid many-objective optimization algorithm for task offloading and resource allocation in multiserver mobile edge computing networks," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3101–3114, Sep./Oct. 2023.

[25] W. Kong, X. Li, L. Hou, J. Yuan, Y. Gao, and S. Yu, "A reliable and efficient task offloading strategy based on multifeedback trust mechanism for IoT edge computing," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13927–13941, Aug. 2022.

[26] G. Rjoub, J. Bentahar, and O. A. Wahab, "BigTrustScheduling: Trust-aware big data task scheduling approach in cloud computing environments," *Future Gener. Comput. Syst.*, vol. 110, pp. 1079–1097, Sep. 2020.

[27] Y. Gong, F. Hao, L. Wang, L. Zhao, and G. Min, "A socially-aware dependent tasks offloading strategy in mobile edge computing," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 3, pp. 328–342, Jul.–Sep. 2023.

[28] T. D. Dang, D. Hoang, and D. N. Nguyen, "Trust-based scheduling framework for big data processing with MapReduce," *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 279–293, Jan./Feb. 2022.

[29] C. Ge, L. Zhou, G. P. Hancke, and C. Su, "A provenance-aware distributed trust model for resilient unmanned aerial vehicle networks," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12481–12489, Aug. 2021.

[30] S. Huang, Z. Zeng, K. Ota, M. Dong, T. Wang, and N. N. Xiong, "An intelligent collaboration trust interconnections system for mobile information control in ubiquitous 5G networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 347–365, Jan.–Mar. 2021.

[31] M. Huang, Z. Li, F. Xiao, S. Long, and A. Liu, "Trust mechanism-based multitier computing system for service-oriented edge-cloud networks," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 4, pp. 1639–1651, Jul./Aug. 2024.

[32] R. Chen and X. Wang, "Maximization of value of service for mobile collaborative computing through situation-aware task offloading," *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 1049–1065, Feb. 2023.

[33] B. Li, X. Wang, Y. Xin, and E. Au, "Value of service maximization in integrated localization and communication system through joint resource allocation," *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4957–4971, Aug. 2023.
[34] M. Liwang, Z. Gao, and X. Wang, "Let's trade in the future! A futures-enabled fast resource trading mechanism in edge computing-assisted UAV networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3252–3270, Nov. 2021.
[35] M. Liwang and X. Wang, "Overbooking-empowered computing resource provisioning in cloud-aided mobile edge networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 5, pp. 2289–2303, Oct. 2022.
[36] X. Zhu, Y. Li, D. Jin, and J. Lu, "Contact-aware optimal resource allocation for mobile data offloading in opportunistic vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7384–7399, Aug. 2017.
[37] X. Wen, J. Chen, Z. Hu, and Z. Lu, "A p-opportunistic channel access scheme for interference mitigation between V2V and V2I communications," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3706–3718, May 2020.
[38] M. Ghorbanzadeh, A. Abdelhadi, and C. Clancy, "Application-aware resource allocation of hybrid traffic in cellular networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 2, pp. 226–241, Jun. 2017.
[39] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.
[40] R. Ding, F. Gao, and X. S. Shen, "3-D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7796–7809, Dec. 2020.
[41] C. Qiu, X. Wang, W. Shen, and R. Lee, "Dynamic construction and adaptation of 3-D virtual network topology for UAV-assisted data collection," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM WKSHPS)*, 2023, pp. 1–6.
[42] H. Fang, X. Wang, and L. Hanzo, "Adaptive trust management for soft authentication and progressive authorization relying on physical layer attributes," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2607–2620, Apr. 2020.
[43] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu, "Energy-efficient admission of delay-sensitive tasks for mobile edge computing," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2603–2616, Jun. 2018.
[44] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via Markov decision process in mobile-edge computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2483–2493, Feb. 2021.
[45] D.-T. Dinh, T. Fujinami, and V.-N. Huynh, "Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient," in *Proc. 20th Int. Symp. Knowl. Syst. Sci*, 2019, pp. 1–17.
[46] X. Liu, J. Yu, J. Wang, and Y. Gao, "Resource allocation with edge computing in IoT networks via machine learning," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3415–3426, Apr. 2020.
[47] T. Cao, J. Yi, X. Wang, H. Xiao, and C. Xu, "Interaction trust-driven data distribution for vehicle social networks: A matching theory approach," *IEEE Trans. Comput. Soc.*, vol. 11, no. 3, pp. 4071–4086, Jun. 2024.
[48] N. Yang, C. Tang, Z. Xiong, and D. He, "RCME: A reputation incentive committee consensus-based for matchmaking encryption in IoT Healthcare," *IEEE Trans. Services Comput.*, vol. 17, no. 5, pp. 2790–2806, Sep./Oct. 2024.
[49] J. Liu, J. Ren, Y. Zhang, X. Peng, Y. Zhang, and Y. Yang, "Efficient dependent task offloading for multiple applications in MEC-cloud system," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2147–2162, Apr. 2023.

Jiazhi Chen (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Western University, London, ON, Canada.

She worked as a Research Assistant with the Department of Electrical and Computer Engineering, Western University. Her current research interests include communication and network security, trust modeling and management, mobile collaborative computing, task offloading, and machine learning.

Xianbin Wang (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2001.

He is a Professor and a Tier-1 Canada Research Chair of 5G and Wireless IoT Communications with Western University, London, ON, Canada. Prior to joining Western University, he was with the Communications Research Centre Canada, Ottawa, ON, Canada, as a Research Scientist/Senior Research Scientist from 2002 to 2007. From 2001 to 2002, he was a System Designer with STMicroelectronics, Geneva, Switzerland. He has over 600 highly cited journals and conference papers, in addition to over 30 granted and pending patents and several standard contributions. His current research interests include 5G/6G technologies, Internet of Things, machine learning, communications security, and intelligent communications.

Dr. Wang has received many prestigious awards and recognitions, including the IEEE Canada R. A. Fessenden Award, the Canada Research Chair, the Engineering Research Excellence Award at Western University, the Canadian Federal Government Public Service Award, the Ontario Early Researcher Award, and nine best paper awards. He is currently a member of the Senate, Senate Committee on Academic Policy and Senate Committee on University Planning at Western. He also serves on NSERC Discovery Grant Review Panel for Computer Science. He has been involved in many flagship conferences, including GLOBECOM, ICC, VTC, PIMRC, WCNC, CCECE, and ICNC, in different roles, such as General Chair, TPC Chair, Symposium Chair, Tutorial Instructor, Track Chair, Session Chair, and Keynote Speaker. He serves/has served as the editor-in-chief, associate editor-in-chief, and editor/associate editor for over ten journals. He was the Chair of the IEEE ComSoc Signal Processing and Computing for Communications Technical Committee and is currently serving as the Central Area Chair of IEEE Canada. He is a Fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks.

Dr. Shen received the West Lake Friendship Award from Zhejiang Province in 2023, the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory in 2021, the R. A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He serves/served as the General Chair for the 6G Global Conference'23, and ACM Mobihoc'15, and the Technical Program Committee Chair/Co-Chair for IEEE Globecom'24, 16, and 07, IEEE Infocom'14, and IEEE VTC'10 Fall. He is the Past President of the IEEE ComSoc, the Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of the IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and an Engineering Academy of Japan International Fellow.