

Article

Not peer-reviewed version

Service Chain-driven Communication and Computing Integration Networking: A Case Study of Levee Piping Hazard Inspection via Remote Sensing

[Jing Chen](#) , [Lyuzhou Gao](#) ^{*} , [Hongquan Sun](#) , [Siqun Yang](#) , [Zhonggen Wang](#) , [Yuting Wan](#) , Kedi Wang

Posted Date: 12 May 2025

doi: 10.20944/preprints202505.0868.v1

Keywords: Integration networking architecture; collaborative computation; service chains; remote sensing application; hazard identification



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Service Chain-Driven Communication and Computing Integration Networking: A Case Study of Levee Piping Hazard Inspection via Remote Sensing

Jing Chen ^{1,2} , Lyuzhou Gao ^{1,2,*} , Hongquan Sun ^{1,2}, Siquan Yang ^{1,2}, Zhonggen Wang ^{1,2}, Yuting Wan³ and Kedi Wang⁴

¹ National Institute of Natural Hazards, Ministry of Emergency Management of the People's Republic of China, Beijing 100085, China

² Key Laboratory of Compound and Chained Natural Hazards Dynamics, Beijing 100085, China

³ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

⁴ School of Safety Science, Tsinghua University, Beijing 100084, China

* Correspondence: gaolvzhou@ninhm.ac.cn.

Abstract: The deployment of artificial intelligence technology in various emerging network applications has spawned a large number of computing tasks, such as on-the-fly image rendering, which requires dynamic collaboration of multi-dimensional resources from the perspective of communication and computing to meet service requirements. In this paper, we introduce a communication and computation integrated network architecture, referred to as (Com)²INet, which has multi-dimensional, multi-layer and heterogeneous characteristics. The architecture integrates software defined network (SDN) and network function virtualization (NFV) technologies to ensure quality of service (QoS) and quality of experience (QoE). Customized advanced computing services can be implemented as service chains (SCs), which consist of ordered virtual network functions and can be scheduled across domains of end-edge-cloud, space-air-ground, and multiple data centers to facilitate ubiquitous network connectivity and collaborative computation. Moreover, we outline key technologies such as measurement and modeling of multi-dimensional heterogeneous resources, multi-path transmission, heterogeneous resource scheduling, and fault tolerance. Furthermore, an intelligent scheduling scheme with dynamic coordination of multi-dimensional resources is proposed for massive remote sensing images, in which an SC-based stepwise adaptive clustering method is utilized to make the optimal joint decision for levee piping hazard identification. Finally, we explore open issues that underlie efficient collaborative computation in (Com)²INet.

Keywords: integration networking architecture; collaborative computation; service chains; remote sensing application; hazard identification

1. Introduction

With the advent of the era of intelligent everything, various new application scenarios are constantly emerging, such as augmented reality, ultra-high-speed railway, holographic communication, and smart industrial Internet of Things [1]. A large number of heterogeneous terminal devices are connected to the Internet, generating tremendous data at the edge of the network, which needs to be efficiently transmitted and processed [2]. According to the Internet Data Center, by 2025, the annual data generated worldwide will grow to 175 Zettabytes [3]. By leveraging the massive amount of increased data, various advanced services with customized requirements can be provided to enhance the comprehensive analysis capacities of complex application scenarios. For instance, in the scenario of ultra-high-speed railway, the obtained audio and video data are used to effectively ensure the operation safety of ultra-high-speed railway vehicles through artificial intelligence (AI) algorithms. In the scenario of emergency response, the remote sensing data are used to improve rescue efficiency

through AI algorithms. These services have the common characteristics of being compute-intensive and time-sensitive. Therefore, how to promote efficient computing is particularly critical to support plenty of advanced services in the intelligent everything scenarios.

New computing paradigms have been proposed to achieve the above target. Driven by powerful computing capabilities, cloud computing with large storage capacity and high data throughput has become the main platform for the application of AI algorithms [4]. However, this “in-network computing” paradigm has problems such as extended decision transmission delay and increased network burden when interacting with terminal devices. By sinking computing and storage resources from the cloud to edge of the network, edge computing paradigm and fog computing paradigm have arisen, reducing task transmission delay and greatly promotes the development of AI technology [5,6]. With limited computing resources, the compute-intensive tasks are offloaded to multiple edge nodes to collaboratively utilize decentralized computing resources, leading to data privacy issues. To this end, the distributed computing [7], such as federated learning, has attracted widespread attention from industry and academia. As advanced services tend to be customized, diversified, and intelligent, a single computing paradigm is difficult to meet the increasingly complex quality of service (QoS) and quality of experience (QoE) demands. Therefore, it is crucial to build a unified computing platform that integrates multiple computing paradigms, in which massive concurrent computing tasks can be globally orchestrated more efficiency to guarantee advanced service requirements.

Orchestrating massive concurrent computing services on a unified computing platform faces the following challenges. First, orchestrating computing tasks requires dynamic scheduling of multi-dimensional heterogeneous resources from the perspectives of communication and computing. Moreover, considering the upper-layer diversity advanced services and underlying differentiated networks, the computing power and network in the existing networks are relatively independent. In addition, the unified computing platform needs to manage massive concurrent computing tasks generated by various advanced services such as live image analytics of remote sensing and large-scale AI model training, which makes it more complicated to flexibly schedule multi-dimensional heterogeneous resources to handle massive concurrent computing services. Therefore, it is necessary to design a unified computing platform to support customized services by coordinating the scheduling of multi-dimensional heterogeneous resources from the perspective of both communication and computation. Then, diversified computing services are giving rise to masses of QoS and QoE requirements, such as deterministic delay, model accuracy, clarity level, privacy protection, and so forth [8]. Therefore, it is necessary to design an intelligent resource scheduling algorithm to adaptively schedule multi-dimensional heterogeneous resources on the unified computing platform.

On one hand, a novel communication and computation integrated network architecture, referred to as (Com)²INet, is proposed for intelligent scheduling of multi-dimensional resources in heterogeneous environments. Service-oriented principles and intelligence facilitate the integration of software defined network (SDN) [9], programming protocol independent packet processor (P4) technology [10], and network function virtualization (NFV) [11] into the designed architecture, enabling flexible network management and reduced operating costs. To be specific, SDN decouples data plane and control plane, supporting centralized control and device configuration. The combination of SDN and P4 technology allows network administrators to customize reconfigure network functions of data plane devices for supporting new functions and protocols, thereby offloading some network functions to data plane, such as in-band telemetry [12], load balancing [13], and re-routing [14], to achieve flexible network management. Moreover, With the support of SDN and NFV technology, computing services can implement traditional network functions in the form of virtual network functions (VNFs), and can migrate from dedicated hardware devices to general-purpose devices to reduce dependence on hardware devices. In conclusion, huge computing flows can be scheduled across domains of end-edge-cloud, space-air-ground, and multiple data centers, enabling the deep integration of “computing and network”. On the other hand, we present a case study of an intelligent scheduling scheme with dynamic coordination of multi-dimensional resources for massive remote sensing images, where a

stepwise adaptive clustering approach is used to make the optimal joint decision for levee piping hazard identification.

Our main contributions are summarized below.

- The (Com)²INet integrates SDN, P4 and NFV technologies. It is a service-oriented computing paradigm that can schedule multi-dimensional heterogeneous resources across domains of end-edge-cloud, space-air-ground, and multiple data centers to ensure QoS and QoE requirements of advanced services. (Com)²INet architecture consists of three layers: computing service layer, mapping adaption layer and converged network layer, aiming to achieve the coordination of applications and network resources, and the secure isolation of networks and applications.
- We design a service-centric computing service layer with different types of service identifications. The device-centric converged network layer with storage, forwarding and computing capabilities to process the service locally or forward it to other appropriate devices. The function-centric mapping adaption layer generates orchestration strategies to guide operations of converged network layer, vertically decouples computing service layer from converged network layer, and realizes the coordinated scheduling and intelligent integration of the service level and network level.
- With (Com)²INet as support and QoS requirements as constraints, we design a stepwise adaptive clustering method for levee piping hazard identification based on temperature field of UAV thermal infrared remote sensing imagery. The method utilizes a progressive approach to iteratively eliminate background regions and refine the target area, enabling the precise localization of piping hazards. Experimental results demonstrate that the proposed method outperforms benchmarks overall performance in terms of accuracy of target extraction, and computational time of algorithm.

The rest of the article is organized as follows. First, we present the (Com)²INet architecture in the “(Com)²INet: Communication and Computation Integrated Network Architecture” section. Following that we present an application use case, in which the experimental results are given and discussed in the “Use Case : (Com)²INet for Levee Piping Hazard Inspection via Remote Sensing” section. Furthermore, we elaborate on the directions and issues for future research in the “Open Issues” section. Finally, we conclude our work in the “Conclusions” section.

2. (Com)²INet: Communication and Computation Integrated Network Architecture

2.1. System Architecture

As mentioned above, how to meet customized QoS and QoE requirements of advanced computing services remains a challenging problem [15]. To this end, a novel system architecture integrating multi-dimensional resources and heterogeneous networks, abbreviated as (Com)²INet, is proposed as shown in Figure 1. The architecture aims to integrate ubiquitous connectivity and collaborative computation, jointly scheduling heterogeneous resources from both communication and computation perspectives to solve the dilemma that emerging applications require more computing power while available computing power is ubiquitous but not easy to use.

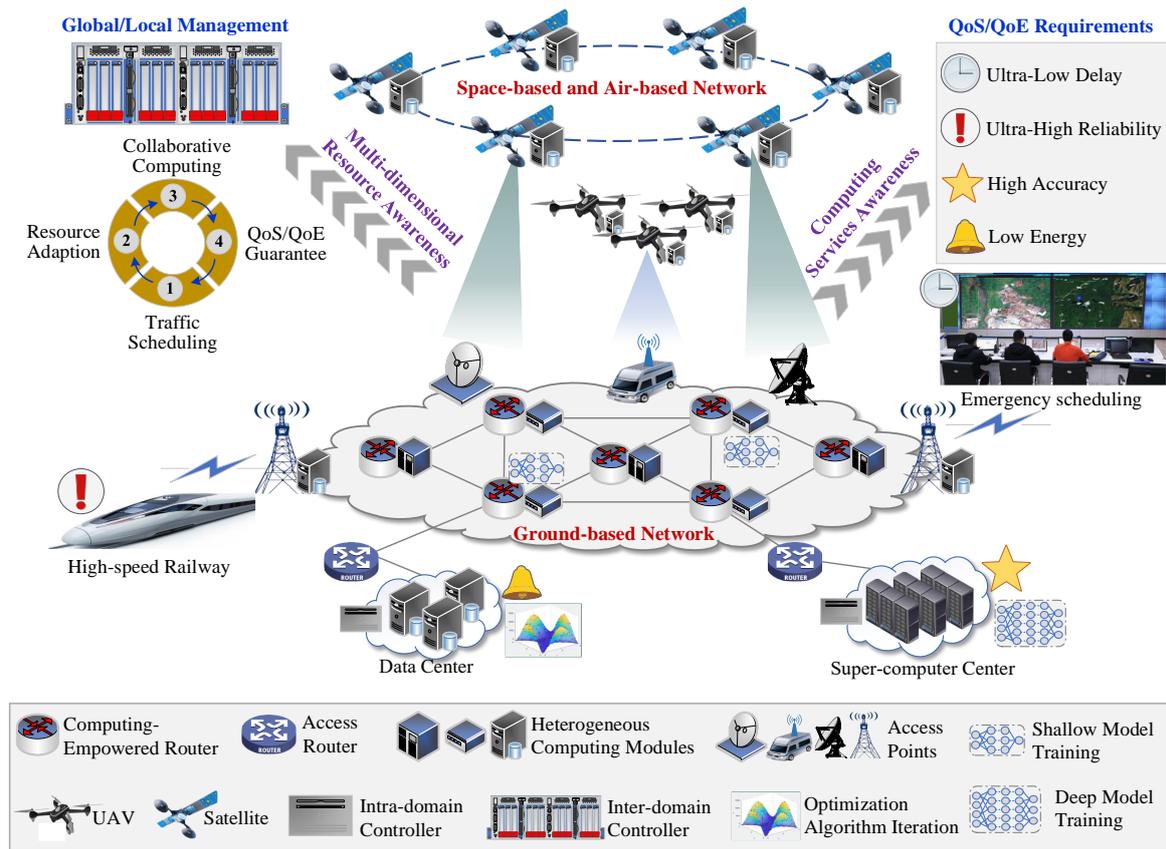


Figure 1. Illustration of the designed $(Com)^2INet$.

$(Com)^2INet$ network has the following new features compared with the traditional forwarding dedicated networks. (1) Heterogeneous network components such as router devices, end devices, edge servers, cloud servers, unmanned aerial vehicle (UAV) and satellites, etc. are equipped with computing units to handle massive concurrent computing tasks [16]. In addition, the designed architecture includes multiple computing centers such as data centers and supercomputer centers [17]. Therefore, massive concurrent flows can be scheduled across domains of end-edge-cloud, space-air-ground, and multiple data centers, support the deep integration of “computing and network” to efficiently satisfy complex QoS and QoE requirements of customized advanced computing services. (2) A hierarchical control mechanism is adopted to jointly perform global and local management. To be specific, the controllers are deployed in each domain to achieve dynamically adjust the network behaviors and manage multi-dimensional network resources. The master controller as a cross-domain orchestrator to centrally control the entire $(Com)^2INet$. It can communicate with other controllers in each domain through the eastbound/westbound interface to exchange aggregated control information [18].

The hierarchical control mechanism divides the functional architecture into three layers, in terms of computing service layer, mapping adaption layer and converged network layer, as shown in Figure 2. We next describe each layer in the $(Com)^2INet$ functional architecture in detail as below.

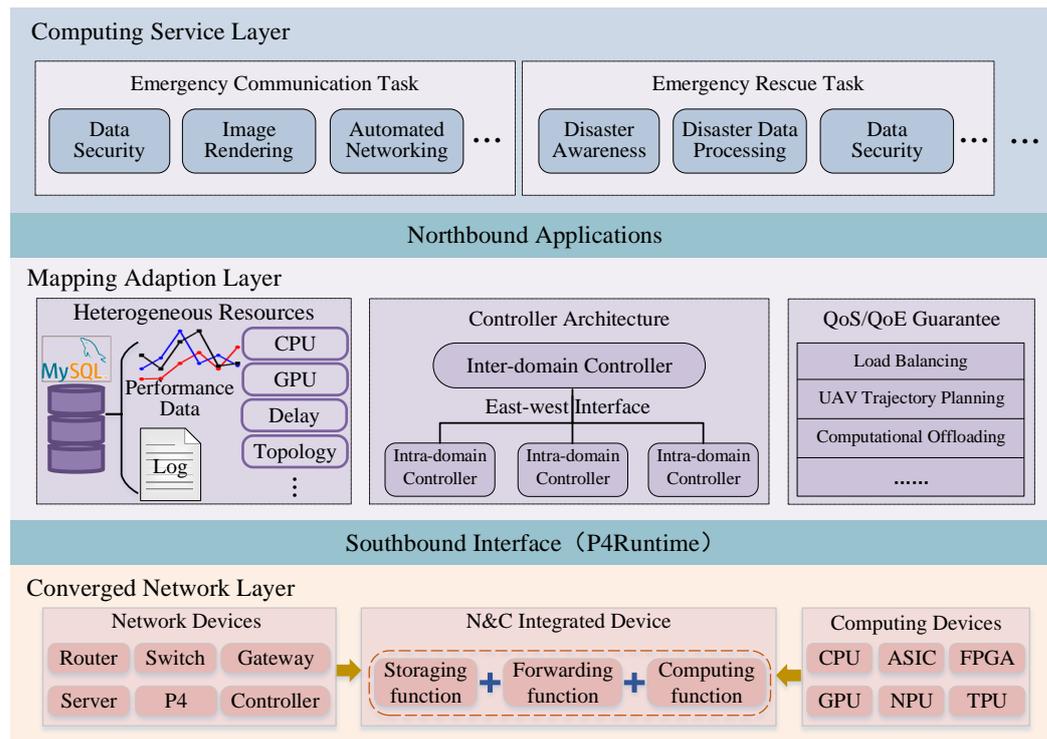


Figure 2. Functional architecture of the designed (Com)²INet.

Computing Service Layer takes service as the basic unit. The attribute characteristics of customized computing services can be identified according to their QoS and QoE requirements to obtain useful information, such as whether the task needs to be split and offloaded to multiple computing devices for processing, how many computing devices need to be scheduled, and which optimal devices to schedule to complete the task. The above critical information can provide service-level support for the integrated scheduling of “computing and network” [19]. In addition, the computing tasks can be divided into different types, such as emergency communication tasks and emergency rescue tasks, and further divided into general computing tasks and intelligent computing tasks to facilitate global task management. Generally, computational process of general computing tasks with models that require global consistency or data integrity, such as linear regression [20] and K-means [21], should be executed on an edge server with required mathematical tool installed and cannot be split and need to be performed on separate computing devices. Besides, the computation process of intelligent computing tasks with AI models that require privacy protection or distributed computing, such as federated learning [22] and deep neural network [23], can be split and coordinated among multiple computing devices.

Converged Network Layer takes network component as the basic unit and is mainly responsible for data storage, forwarding and computing, including network devices, computing devices and integrated devices. To be specific, network devices such as switches, routers and gateways constitute the core network and are mainly responsible for data storage and forwarding. The computing devices are composed of one or more processing units such as central processing unit (CPU), network processing unit (NPU), field programmable gate array (FPGA) and graphics processing unit (GPU), which are responsible for processing real-time tasks [24]. Computing devices are distributed at the edge of the network, including multiple computing centers, compared with network devices. Integrated devices integrate computing, storage, and forwarding functions. For instance, router devices can have both storage, forwarding and computing functions by introducing computing function. When the computing-enhanced router device receives a service, it can process the service locally or forward it to other appropriate devices. Note that network components support heterogeneous network protocol communications including IPv4, IPv6, etc, enabling cross-protocol collaborative transmission mechanism.

Mapping Adaption Layer takes the function as the basic unit and is mainly responsible for scheduling multi-dimensional resources. There are three computing paradigms that support the implementation of computing services, in terms of in-network computing [25], out-of-network computing [26] and end-edge-cloud collaborative computing [27]. In-network computing can directly execute lightweight computing tasks in network devices, such as switches and routers, along the task transmission path, instead of transmitting data to traditional computing nodes, such as servers and cloud platforms. Out-of-network computing, with powerful computing capabilities, can orchestrate heavy-weight computing tasks to data centers and supercomputer centers for execution. Moreover, a computing task can be decomposed into multiple micro-services, or a monolithic function can be divided into multiple portions to orchestrate different parts between ends, edges and clouds. For instance, the deep neural network (DNN) model can be partitioned, where the shallow layers are deployed and performed on the ends, the middle layers are deployed and performed on edges, and the other layers are deployed and performed on the cloud.

In summary, the core of the above computing paradigm is to establish relationships between service, group and component. To this end, the service-group mapping and group-component mapping are constructed. According to the customized task requirements and multi-dimensional resource attributes perceived at a fine-grained level, functional modules in terms of inter-domain scheduling, intra-domain scheduling, QoS and QoE guarantee are constructed to achieve dynamic service-component mapping. To be specific, the service-group mapping mechanism is responsible for perceiving and translating the requirements of massive concurrent computing tasks, and establishing a service group at the logical level to promote the coordinated processing of each computing task. The group-component mapping mechanism is responsible for introducing established service group into the physical network, aggregating network components to coordinate and schedule heterogeneous resources to accomplish computing tasks.

(Com)²INet architecture decouples computing service layer from converged network layer vertically through the mapping adaptation layer, realizing the coordinated scheduling and intelligent integration of the service level and network level. On one hand, (Com)²INet architecture is conducive to network providing users with customizable computing services, and on the other hand, it is also conducive to providing users with virtual private networks to satisfy the requirements of related computing services in terms of reliability, security, and scalability. Furthermore, (Com)²INet architecture horizontally decouples the control plane and the operation plane. The upper application plane communicates with the control plane through the northbound interface, and the underlying data plane communicates with the control plane through the southbound interface (e.g., P4Runtime). The control plane uniformly schedules heterogeneous network resources, translates decisions into configuration files and sends them to data plane to complete computing services.

2.2. Key Features of (Com)²INet Computational Paradigm

Service-centric paradigm. In (Com)²INet computational paradigm, services become the core components and driving force of the network. The traditional network architecture is centered on data transmission. In (Com)²INet architecture, communication and computation are deeply integrated, and services need determine the allocation and scheduling of multi-dimensional resources. With the help of NFV technology, computing services are translated into corresponding service chains (SCs), and end-to-end services are realized by sequentially connecting the required network functions to satisfy the diverse requirements of communication and computation [28].

Serverless-based model. Network functions can be chained in an event-driven manner, typically initialized within milliseconds [29]. This on-demand chaining model enables the system to orchestrate different locations and differentiated resources in real time to achieve the optimal combination of user experience, resource utilization and network efficiency. In addition, the model can easily update and replace specific network functions to adapt to new services or improve algorithm performance, enabling an efficient, intelligent, and elastic management paradigm.

Regionless-based model. In $(Com)^2INet$ architecture, heterogeneous network components are identified by names of network functions instead of IP addresses. $(Com)^2INet$ jointly considers heterogeneous resources from both the communication and computation perspectives, and schedules customized services to target network components for computing and analysis along optimal paths in a decentralized manner, achieving collaborative scheduling within the WAN. Regionless-based model solves the “IP bottleneck” problem and enables the application of new network architecture and service-centric paradigm [30].

In this paper, we propose a new computing paradigm, named $(Com)^2INet$, which jointly schedules heterogeneous resources from both communication and computation perspectives to solve the dilemma that emerging applications require more computing power while available computing power is ubiquitous but not easy to use.

2.3. Key Technologies

(1) *Measurement and modeling of multi-dimensional heterogeneous resources.* The computing capability of network component n , referred to as CC_n , is measured by indicators such as FLOPS, IOPS, BOPS and LOPS, etc., so as to satisfy the differentiated calculation rate requirements of complex and diversified services. The mathematical expressed as $CC_n = \{FLOPS_n, IOPS_n, BOPS_n, LOPS_n, \dots\}$. The storing capability of network component n , referred to as SC_n , is measured by indicators such as disk capacity DC , disk bandwidth DB , Input/Output Per Second $IOPS$, memory capacity MC , memory bandwidth MB , etc. The mathematical expressed as $SC_n = \{DC_n, DB_n, IOPS_n, MC_n, MB_n, \dots\}$. The networking capability of network link l is measured by indicators such as bandwidth BW , end-to-end delay D , delay jitter DJ , packet loss rate PL , throughput TP , etc. The mathematical expressed as $NC_l = \{BW_l, D_l, DJ_l, PL_l, TP_l, \dots\}$.

(2) *Multi-path transmission.* In the $(Com)^2INet$ computational paradigm, multi-path transmission mechanism has the characteristics of dynamic awareness, resource coordination and efficient utilization, and supports data parallelism and task splitting, as described in Figure 3. By perceiving task requirements, network environment and distribution of multi-dimensional resources, and comprehensively considering factors such as network load, traffic type, and delay constraints, it intelligently selects multiple optimal paths to achieve cross-regional, multi-link adaptive transmission. This mechanism can avoid congestion on a single path, improve data transmission efficiency, and improve multi-dimensional resource utilization.

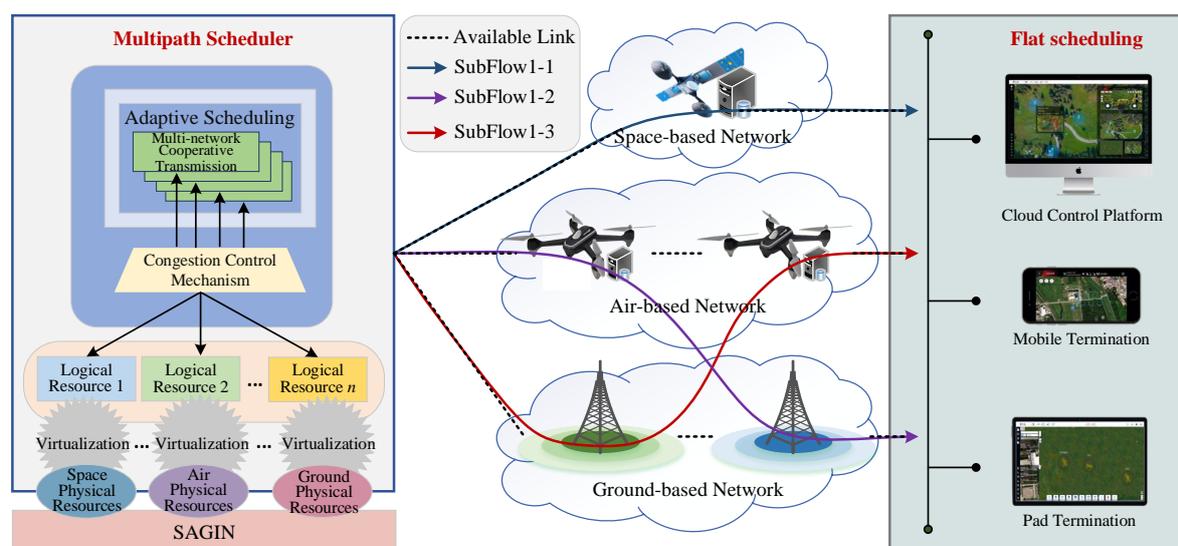


Figure 3. Illustration of multi-path transmission.

(3) *Heterogeneous resource scheduling.* In the $(Com)^2INet$ architecture, heterogeneous resource scheduling supports scheduling of massive concurrent computing tasks across multiple computing

domains such as end-edge-cloud, space-air-ground, and multiple data centers, as shown in Figure 4. It is worth noting that two or more domains can be selected simultaneously to effectively coordinate computing, storage, and network resources with different geographical distributions to provide computing services for tasks with extremely strict QoS and QoE requirements. Heterogeneous resource scheduling requires comprehensive consideration of task characteristics (i.e., parallel mode of large model training), traffic patterns (i.e., mixed long and short flows, burst flows), network topology and other factors, and achieves global optimal resource adaptation and computing task execution efficiency by building an efficient, flexible and intelligent adaptive optimization form.

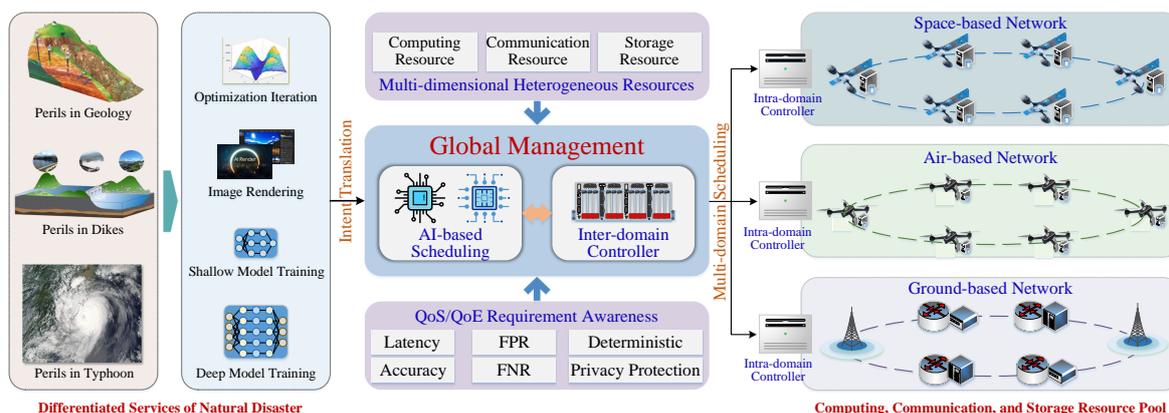


Figure 4. Illustration of heterogeneous resource scheduling.

(4) *Fault tolerance.* In the (Com)²INet architecture, fault-tolerance mechanism is designed to ensure that in a complex network environment, when faced with the network component/link failure, overload or other sudden abnormal situations, the abnormal point can be quickly located through the intelligent fault detection and diagnosis system. The mechanism comprehensively considers multi-dimensional factors such as network topology, resource utilization, QoS and QoE requirements, and based on preset fault tolerance strategies and real-time network status, dynamically performs operations such as intelligent re-routing, dynamic load balancing, or flexible migration of network functions to ensure the continuity of data transmission, uninterrupted critical services, and undegraded service quality.

As illustrated in Figure 5, the computing services are translated into SCs, routing from ingress node s to egress node d , and through the ordered network functions $\{v_1, v_2, \dots, v_i\}$ required for the computing services. For example, when UAV u_1 fails for some reason, the fault tolerance system makes the migration decision and migrates the network function v_1 hosted on u_1 to UAV u_2 that satisfies the network constraints. The link is rerouted from $\{m_4 - u_1 - n_1 - n_2\}$ to $\{m_4 - u_2 - n_1 - n_2\}$.

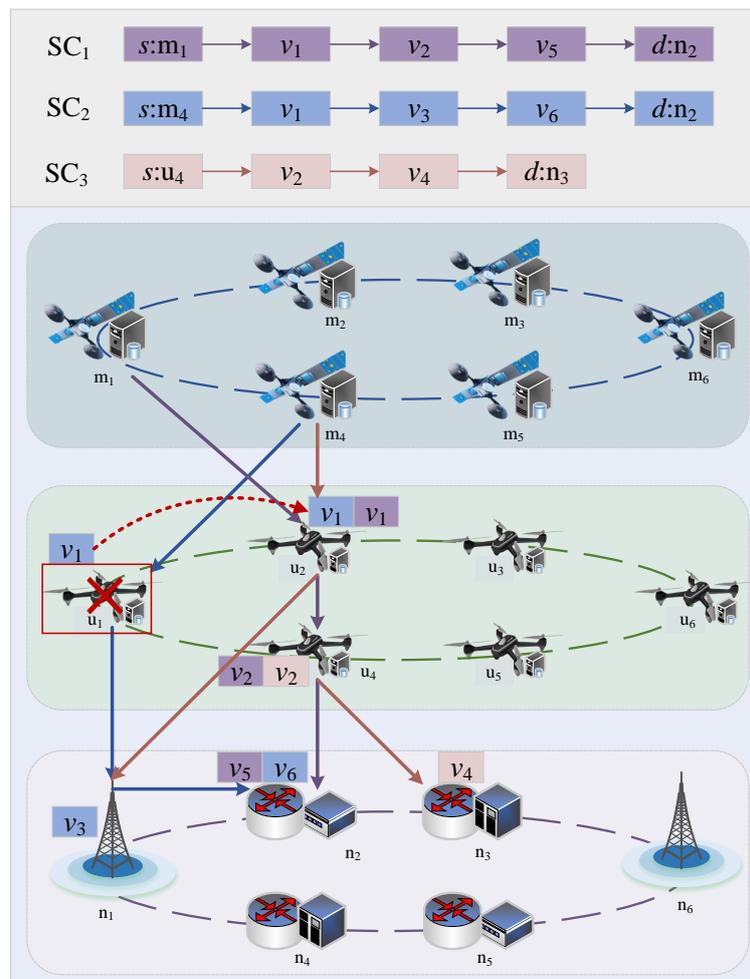


Figure 5. Illustration of fault tolerance.

2.4. Working Principle

The working principle for implementing computing services in $(Com)^2INet$ architecture is depicted in Figure 6. The environment awareness module perceives computing service requirements, network functions, and multi-dimensional heterogeneous resources from both the communication and computation perspectives. To be specific, computing services measurement translates computing service requirements into actually required heterogeneous resources and customized network functions through intent translation to improve the efficiency of interaction with users. The network function measurement is to measure the heterogeneous resources required for different functions, so as to effectively understand the heterogeneous resources required for computing services call functions. The perceived translated services and heterogeneous resources are notified to scheduling problem modeling serverless platform for modeling multi-objective joint optimization with multi-network constraints. Customized network functions are chained, executed, and billed according to current services. In specific, serverless platform builds SCs for arrival services, and SCs schedule resources in the heterogeneous infrastructure pooling.

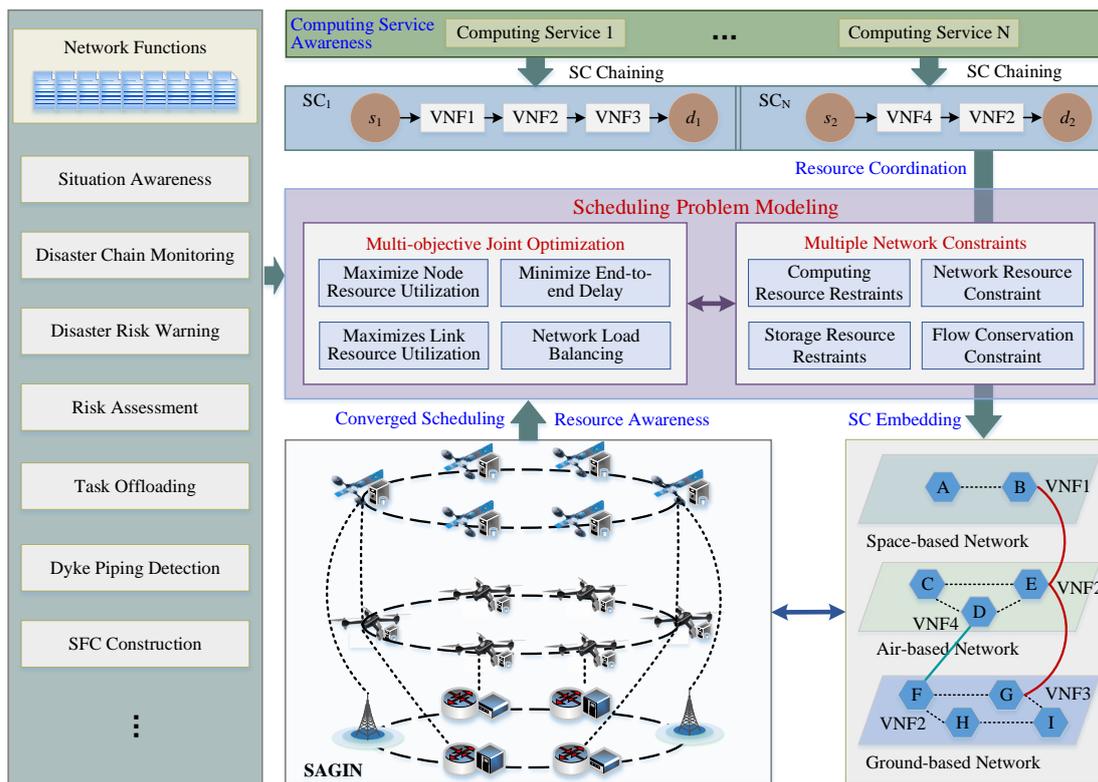


Figure 6. Working principle for implementing computing services in $(Com)^2INet$ architecture.

During orchestrating phase, we establish scheduling mechanism from three perspectives: users, networks and providers. First, the scheduling mechanism should adaptively satisfy various QoS requirements from user's perspective, in terms of delay requirement, heterogeneous resources requirement and ultra-reliable transmission, and so forth. Then, $(Com)^2INet$ infrastructure integrates awareness, forwarding, storage, computing and processing capabilities, which puts forward greater flexibility requirements for network management. Moreover, an effective resource scheduling mechanism should enable the provider to obtain the maximum benefit to motivate more providers to participate in the communication and computation integrated $(Com)^2INet$. Furthermore, the obtained scheduling mechanism is translated into the network configuration files, which guide the optimal network components to host required SCs to accomplish the computing service in serial or parallel manner.

In summary, $(Com)^2INet$ can realize the mapping from computing service to a set of network components through the intelligent integration of multi-space and multi-dimensional heterogeneous resources, so as to facilitate dynamic and on-demand adjustment of services and networks.

3. Use Case: $(Com)^2INet$ for Levee Piping Hazard Inspection via Remote Sensing

Piping leakage is a critical factor leading to levee breaches and catastrophic flooding disasters. Levee breaches often result in mass casualties, making piping leakage a major threat to flood disaster prevention and mitigation in China. According to statistics, China's river levees extend over 328,000 kilometers, protecting a population of 650 million and 42 million hectares of farmland. However, more than 90% of these river levees consist of aging earth-rock dams, which are highly susceptible to severe hazards during the annual flood season due to high water levels. UAV-based remote sensing technology has demonstrated significant potential in the inspection and identification of piping hazards in levees. UAV platforms are characterized by their portability, lightweight design, and low cost, enabling rapid acquisition of levee data. Furthermore, various sensors can capture diverse morphological data of levees, providing abundant information sources for identifying piping hazards. The remote sensing

data processing techniques designed based on the (Com)²INet architecture can achieve rapid, accurate, and large-scale inspections of levee hazards.

However, current UAV-based remote sensing approaches for levee piping hazard inspection face challenges such as low recognition accuracy and poor scalability. Although piping hazards are visually prominent in thermal infrared imagery, such imagery is heavily influenced by weather conditions. The piping features extracted from thermal infrared imagery exhibit significant variability under different weather and environmental conditions, leading to poor stability in distinguishing piping hazards from other surface objects. Moreover, the complex flood-season levee environment introduces interference to recognition algorithms. The levee surface is covered by diverse natural and manmade objects, and the harsh conditions during the flood season further complicate the scene. Particularly in large-scale, multi-sensor, and multi-object application scenarios, uncertain objects often affect the performance of recognition algorithms.

Currently, deep learning-based object recognition algorithms typically rely on large sample datasets. However, due to the difficulty in obtaining piping hazard data, the demand for extensive training samples cannot be met. This study proposes an unsupervised machine learning-based method, SC-based stepwise adaptive clustering method (SACM), for piping hazard segmentation using UAV thermal infrared imagery. The SC-based SACM method extracts piping hazards through iterative clustering and adaptive strategies. First, thermal infrared imagery is converted into temperature maps. Second, a series of binary classifications is applied to the temperature maps to progressively exclude background regions unrelated to piping hazards, thereby mitigating interference caused by complex levee environments. Finally, the elbow method is introduced to determine the hyperparameters of the clustering algorithm, overcoming the dependence on subjective manual settings for cluster centers and achieving adaptive clustering based on data features.

SC-based SACM is an unsupervised clustering algorithm specifically designed for piping hazard segmentation in UAV thermal infrared imagery. Unlike traditional clustering algorithms, SC-based SACM employs a progressive and adaptive strategy to iteratively refine the segmentation process. By leveraging temperature field analysis and stepwise clustering, SC-based SACM effectively eliminates irrelevant background regions and accurately extracts piping hazard targets. This method is particularly robust in complex environments, ensuring high precision and reliability in hazard detection. The SC-based SACM algorithm is composed of three key steps: temperature field preprocessing, background removal, and stepwise adaptive clustering. Each step can be modeled as a VNF and chained into an SC, which can be executed sequentially on multiple computing devices to achieve the scalable levee piping hazard inspection. The SC-based SACM algorithm is designed to overcome challenges such as interference from complex levee environments, variability in thermal infrared imagery due to weather conditions, and the subjectivity of manual parameter selection. By introducing the elbow method for hyperparameter determination, the SC-based SACM achieves fully adaptive clustering based on data characteristics, without requiring predefined cluster centers. The specific process is illustrated as Figure 7, and the main pseudo-code is as follows.

SC-based SACM for levee piping hazard inspection

Input: UAV thermal infrared imagery I ; Convergence threshold ϵ ; Maximum iterations T .

Process:

1. Temperature Field Preparation:

- Convert the thermal infrared imagery I into a temperature map T using the thermal infrared radiative transfer mechanism.
- Normalize the temperature map T to ensure consistent data scaling across different images.

2. Background Removal:

- Perform a series of binary segmentations on the temperature map T to isolate potential piping hazard regions:
 - Calculate the temperature threshold τ for binary segmentation using statistical analysis of the temperature distribution.
 - Segment the image into background regions and suspected target regions based on τ .
- Iteratively refine the segmentation by updating τ to progressively exclude irrelevant background regions.

3. Stepwise Adaptive Clustering:

- Apply adaptive clustering to the refined target regions:
 - Use the elbow method to determine the optimal number of clusters k based on the within-cluster sum of squares (WCSS)
 - Initialize cluster centroids adaptively based on the temperature distribution of the target regions.
 - Perform iterative clustering to assign each data point to the nearest cluster and update cluster centroids.
- Continue the clustering to assign each data point to the nearest cluster and update cluster centroids.

4. Output:

- Extracted piping hazard regions with precise boundaries.
 - Cluster assignments $U = \{u_1, u_2, \dots, u_k\}$ representing the segmented target areas.
-

(i) *Land surface temperature retrieval from UAV thermal infrared imagery.* Utilizing thermal infrared (TIR) imagery acquired by UAVs, the land surface temperature is retrieved according to the thermal infrared radiative transfer mechanism. This step ensures the accurate reconstruction of temperature fields, providing a solid foundation for subsequent analysis.

(ii) *Levee piping temperature field analysis for image background removal.* Based on the retrieved temperature map, the thermal characteristics of piping targets are analyzed. This involves extracting the temperature distribution features of the target region and performing binary segmentation to separate the image into background regions and regions containing potential piping targets. This segmentation step effectively removes irrelevant background areas, reducing interference for subsequent clustering analysis.

(iii) *Feature construction using stepwise adaptive clustering.* Following background removal, stepwise adaptive clustering is employed to construct the thermal field features of the piping targets. This process includes further refinement of the segmented target regions to extract suspected piping areas. Automatic clustering is then applied to further delineate and refine the target regions, avoiding the subjectivity associated with manual threshold selection. The SC-based SACM approach for levee piping hazard inspection is particularly robust and adaptable for extracting piping targets in complex scenarios, ensuring high scalability and accuracy in feature construction.

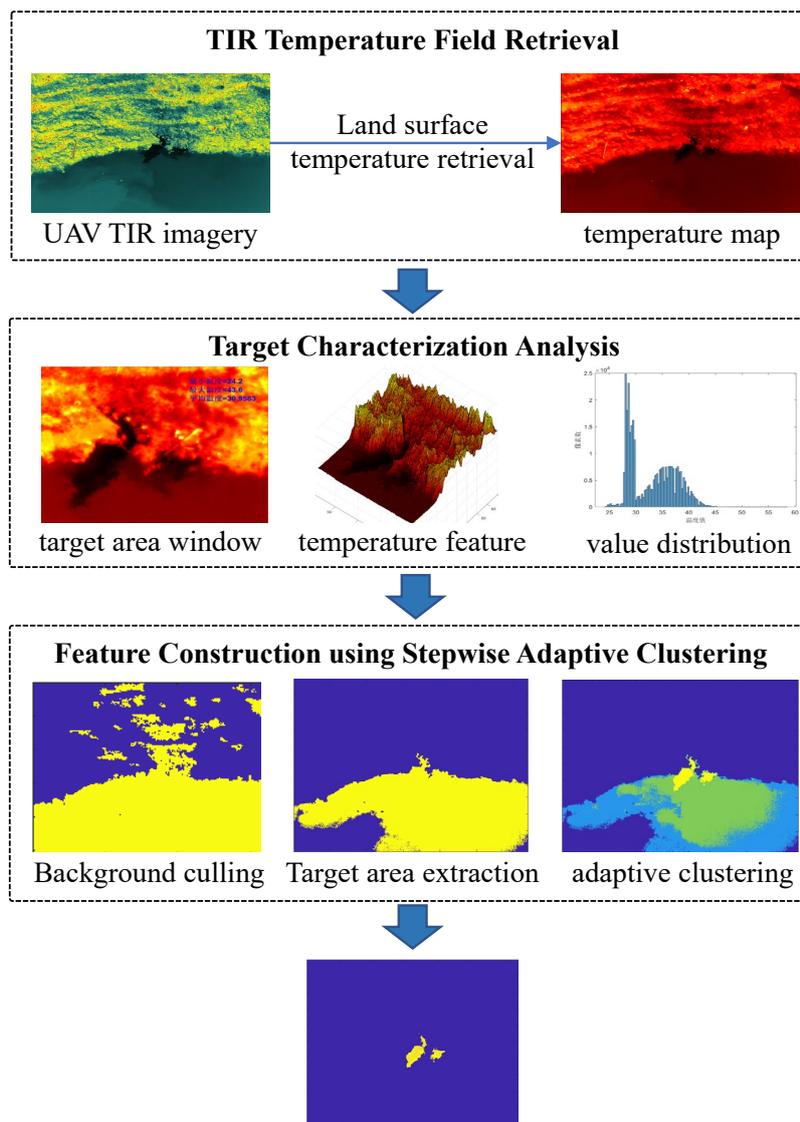


Figure 7. Illustration of SC-based SACM for levee piping hazard inspection via remote sensing.

Benchmarks. (1) *k-Medoids* (KM) algorithm is a classical partitioning clustering algorithm widely used in data mining. Unlike *k-Means*, which relies on centroids as cluster representatives, KM selects actual data points (medoids) to represent clusters, making it more robust to noise and outliers. The algorithm aims to minimize the sum of dissimilarities between data points and their corresponding medoids, ensuring the formation of compact and well-separated clusters. KM algorithm is particularly suitable for scenarios with non-Euclidean distance metrics or when the dataset contains outliers that could significantly distort the results of *k-Means*. By iteratively refining the medoids and cluster assignments, KM seeks to achieve optimal clustering with respect to the chosen dissimilarity measure. Because of the global dependence on the central object, the KM cannot be split and can only be executed as a whole function on a separate computing device. The main pseudo-code is as follows.

KM-based levee piping hazard inspection

Input: Dataset $D = \{x_1, x_2, \dots, x_m\}$; Number of clusters k .

Process:

1. Randomly select k samples from D as the initial centroids $\{\mu_1, \mu_2, \dots, \mu_k\}$.
2. **Repeat:**
 3. Initialize $C_i = \emptyset$ for $1 \leq i \leq k$.
 4. For $j = 1, 2, \dots, m$:
 - Compute the distance between sample x_j and each centroid $\mu_i (1 \leq i \leq k)$: $d_{ji} = \|x_j - \mu_i\|_2$.
 - Assign x_j to the cluster with the nearest centroid: $\lambda_j = \operatorname{argmin}_{i \in \{1, 2, \dots, k\}} d_{ji}$.
 - Update the cluster: $U_{\lambda_j} = U_{\lambda_j} \cup \{x_j\}$.
 - v. **End For**
 - vi. **For** $i = 1, 2, \dots, k$:
 - Compute the new centroid for each cluster: $\mu'_i = \frac{1}{|U_i|} \sum_{x \in U_i} x$.
 - If** $\mu'_i \neq \mu_i$:
 - Update the centroid: $\mu_i = \mu'_i$.
 - Else:**
 - Keep the current centroid unchanged.
 - vii. **End For**
3. **Until** centroids remain unchanged or maximum iterations are reached.

Output: Cluster assignments $U = \{u_1, u_2, \dots, u_k\}$.

(2) *Fuzzy C-Means* (FCM) is a widely used unsupervised clustering algorithm based on fuzzy set theory. Unlike traditional hard clustering methods, such as k-Means, where each data point is assigned to exactly one cluster, FCM allows a data point to belong to multiple clusters with varying degrees of membership. This flexibility makes FCM particularly suitable for applications where data points exhibit overlapping characteristics or uncertainty in cluster boundaries. The objective of FCM is to minimize the weighted sum of squared errors between data points and cluster centers, with the weights determined by the membership degrees. The algorithm iteratively updates cluster centers and membership degrees until convergence. Since FCM requires global consistency, it cannot be split and can only be executed as a whole function on a separate computing device. The main pseudo-code is as follows.

FCM-based levee piping hazard inspection

Input: Dataset $D = \{x_1, x_2, \dots, x_m\}$; Number of clusters k ; Fuzziness parameter $m > 1$; convergence threshold $\epsilon > 0$; Maximum iterations T .

Process:

1. **Initialization:**
 - Randomly initialize the membership matrix $H = [h_{ij}]$, where $h_{ij} \in [0, 1]$ and $\sum_{i=1}^k h_{ij} = 1$ for all j .
2. **Repeat:**
 3. **Update Cluster Centers:**
 - Compute the cluster center c_i for $i = 1, 2, \dots, k$: $c_i = \frac{\sum_{j=1}^m h_{ij} x_j}{\sum_{j=1}^m h_{ij}}$.
 - iv. **Update Membership Degrees:**
 - For each data point x_j and cluster i , update the membership degree h_{ij} :

$$h_{ij} = \frac{1}{\sum_{l=1}^k \left(\frac{\|x_j - c_i\|}{\|x_j - c_l\|} \right)^{\frac{2}{m-1}}}$$
 - v. **Check Convergence:**
 - Compute the change in membership matrix δH . If $\Delta H < \epsilon$ or the maximum number of iterations reached, stop the iteration.
3. **End Repeat.**
4. Assign each data point x_j to the cluster with the highest membership degree.

Output: Cluster centers $U = \{u_1, u_2, \dots, u_k\}$; Membership matrix $H = [h_{ij}]$.

(3) *Gaussian mixture model* (GMM) is a probabilistic model widely used for clustering tasks. It assumes that data points are generated from a mixture of several Gaussian distributions with unknown parameters. Each Gaussian component represents a cluster, and GMM algorithm aims to model the

entire dataset as a weighted sum of these Gaussian distributions. GMM is particularly effective for datasets with overlapping clusters and can model more complex cluster shapes compared to simpler algorithms like k-Means. Unlike hard clustering methods, such as k-Means, which assign each data point to a single cluster, GMM performs soft clustering by assigning each data point a probability of belonging to each cluster. This probabilistic approach makes GMM more flexible in capturing underlying structure of the data. For the soft clustering process of GMM, the posterior probability calculation and parameter update must be based on global data to ensure model convergence and accuracy. Therefore, the GMM model cannot be split and can only be executed as a whole function on a separate computing device. The main pseudo-code is as follows.

GMM-based levee piping hazard inspection

Input: Dataset $D = \{x_1, x_2, \dots, x_m\}$; Number of clusters k ; convergence threshold $\epsilon > 0$.

Process:

1. Initialization:

- Randomly initialize the parameters of the Gaussian components.
 - Means $\chi_i (i = 1, \dots, k)$.
 - Covariance matrices $\Sigma_i (i = 1, \dots, k)$.
 - Mixing coefficients π such that $\sum_{i=1}^k \pi = 1$.

2. Expectation-Maximization (EM) Algorithm:

• **Repeat:**

• **E-step(Expectation):**

- For each data point $x_j (j = 1, \dots, m)$ and each Gaussian component $i (i = 1, \dots, k)$, compute the posterior probability (responsibility) that x_j belongs to the i -th

Gaussian: $\gamma_{ji} = \frac{\pi_i \cdot \mathbb{N}(x_j | \chi_i, \Sigma_i)}{\sum_{l=1}^k \pi_l \cdot \mathbb{N}(x_j | \chi_l, \Sigma_l)}$ where $\mathbb{N}(x_j | \chi_i, \Sigma_i)$ is the Gaussian probability density function: $\mathbb{N}(x_j | \chi_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}}$.

• **M-step(Maximization):**

- Update the parameters of the Gaussian components based on the responsibility γ_{ji} :

a. Update the mixing coefficients: $\pi_i = \frac{1}{m} \sum_{j=1}^m \gamma_{ji}$

b. Update the means: $\chi_i = \frac{\sum_{j=1}^m \gamma_{ji} x_j}{\sum_{j=1}^m \gamma_{ji}}$

c. Update the covariance matrices: $\Sigma_i = \frac{\sum_{j=1}^m \gamma_{ji} (x_j - \chi_i)(x_j - \chi_i)^T}{\sum_{j=1}^m \gamma_{ji}}$

• **Check for Convergence:**

- Compute the log-likelihood of the data under the current model:

$$\log L = \sum_{j=1}^m \log \left(\sum_{i=1}^k \pi_i \cdot \mathbb{N}(x_j | \chi_i, \Sigma_i) \right).$$

- If the change in log-likelihood is less than the threshold ϵ stop the iteration.

3. Output:

- The parameters of the Gaussian components $(\chi_i, \Sigma_i, \pi_i)$ and the cluster assignments based on the maximum posterior probability: Cluster assignment for $x_j = \operatorname{argmax}_i \gamma_{ji}$.
-

Performance Evaluation. *Evaluation Indicator.* Mean Intersection over Union (mIoU) and total runtime of per frame are adopted in this paper for evaluating accuracy of target extraction and algorithm efficiency. The mIoU quantitatively measures the accuracy of predicted segmentation results compared to the label truth. It evaluates the overlap between the predicted segmentation region and the true region for each class and computes the average across all classes. As a robust and interpretable metric, mIoU is particularly suitable for assessing the performance of clustering and segmentation algorithms, including those applied to UAV thermal infrared imagery for levee piping hazard detection. For the specific target, the IoU and the total runtime are computed as below.

$$IoU = \frac{TruePositive}{TruePositive + FalsePositive + FalseNegative} \quad (1)$$

$$T_{total} = T_{end} - T_{start}, \quad (2)$$

where True Positive is the number of pixels correctly predicted as belonging to piping target, False Positive is the number of pixels incorrectly predicted as belonging to piping target. False Negative if the number of pixels belonging to piping target.

Experimental Data. The experimental data in this manuscript were obtained using remote sensing images captured by the DJI H20T sensor mounted on the DJI M300 UAV. The images were taken from a nadir perspective with a flight speed of 8 meters per second, primarily acquiring thermal infrared remote sensing images. The data collection took place in the Pajiang River Basin, Fogang County, Qingyuan City, Guangdong Province, on April 20, 2023. The UAV operated at a flight altitude of 30 meters, and the thermal infrared camera had a resolution of 640×512 with a Noise Equivalent Temperature Difference (NETD) $\leq 50mK@f/1.0$. During this data acquisition, eight frames of thermal infrared remote sensing images containing piping hazards were collected, as shown in Figure 8. These thermal infrared remote sensing images with piping hazards were used in this manuscript to test and evaluate the performance of the proposed SC-based SACM algorithm, which in terms of accuracy of target extraction and computational time of benchmarks.

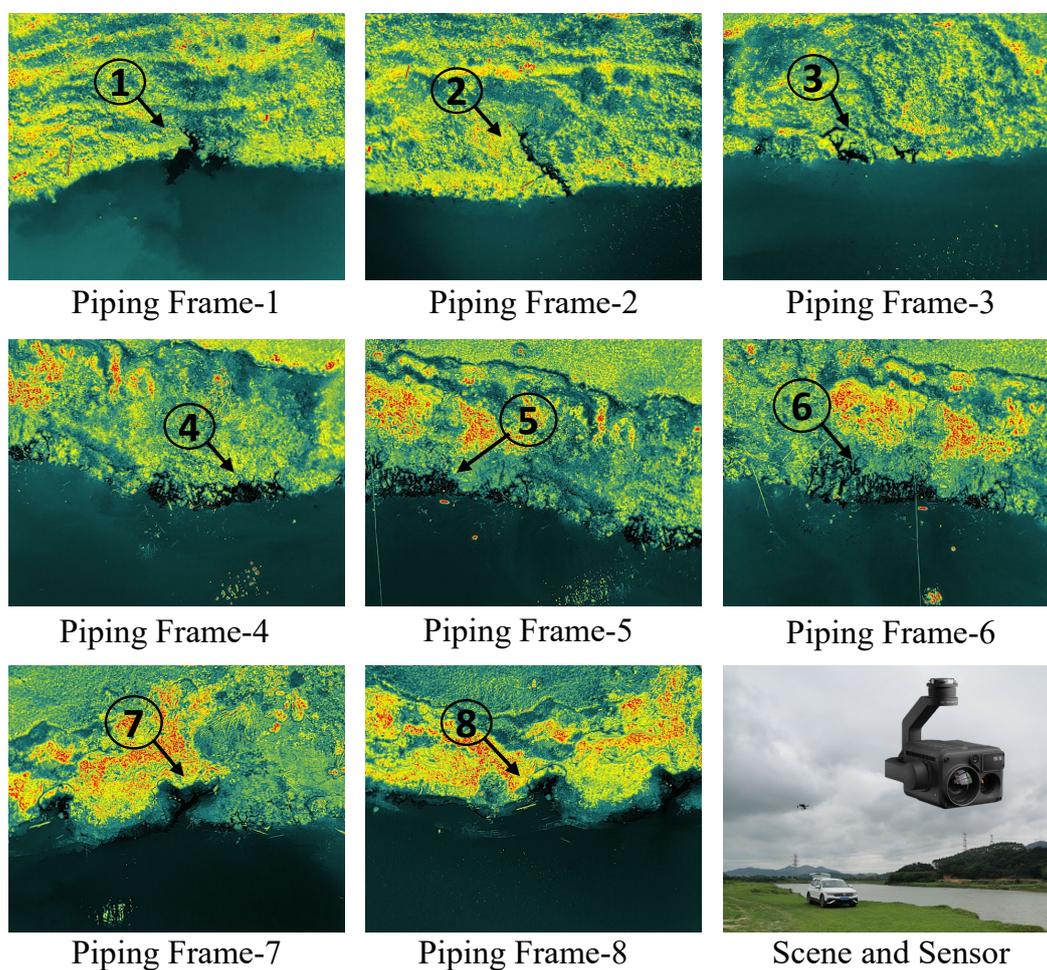


Figure 8. Eight frames of thermal infrared imagery with piping targets.

Comparison in accuracy of target extraction. As shown in Table 1, SACM demonstrates a significant advantage in terms of target extraction accuracy. As indicated by the data of Table 1, SACM consistently achieves the highest mIoU values across all tests, with an average mIoU of 0.1469. This suggests that SACM provides superior precision in extracting piping hazards. In contrast, traditional algorithms such as KM, GMM, and FCM exhibit lower average mIoU values of 0.0423, 0.0269, and 0.0303, respectively. These conventional methods often struggle with the complexities of thermal infrared remote sensing imagery, where background noise and environmental variations can adversely affect target extraction accuracy. SACM effectively mitigates irrelevant background regions through its

progressive adaptive clustering strategy, enhancing the precision of piping hazard extraction. Its robust and reliable performance in complex environments is evident, maintaining stability under varying weather conditions. Consequently, SACM demonstrates superior performance in piping detection tasks, validating its applicability and efficacy in challenging scenarios.

Table 1. Accuracy of target extraction (mIoU).

Piping Number	KM	GMM	FCM	SACM
1	0.0015	0.0266	0.0000	0.4694
2	0.0173	0.0078	0.0019	0.0198
3	0.0157	0.0013	0.0148	0.1094
4	0.0546	0.0600	0.0604	0.1662
5	0.1016	0.0019	0.0908	0.0741
6	0.0712	0.0788	0.0747	0.0755
7	0.0213	0.0182	0.0000	0.0165
8	0.0550	0.0208	0.0000	0.2441
Mean	0.0423	0.0269	0.0303	0.1469

Comparison in computational time. As shown in Table 2, SC-based SACM also exhibits a remarkable advantage in computational efficiency. SC-based SACM achieves the lowest average runtime per frame at 0.2288s, significantly outperforming other algorithms. This indicates that SC-based SACM is highly efficient in processing thermal infrared imagery for piping hazard detection. In contrast, traditional monolithic-based algorithms, such as KM, GMM, and FCM, have average runtimes of 6.7628, 96.3845, and 209.3182 seconds per frame, respectively. To be specific, the time complexity of monolithic-based KM is $O(k(n-k)^2)$, where n indicates the number of data points and k indicates the number of clusters. The time complexity of monolithic-based GMM is $O(nkD^2)$, where D indicates the dimension of the covariance matrix. The time complexity of monolithic-based FCM is $O(nkcI)$, where c indicates the number of iterations and I indicates the dimension of data. Since FCM needs to process fuzzy membership, the calculation process is relatively complicated and usually requires multiple iterations to converge, so the computational time is higher. The efficiency of SC-based SACM is attributed to the fact that the model can be split and coordinated among multiple computing devices, and its streamlined clustering process, which reduces computational overhead while maintaining higher accuracy. This efficiency makes SC-based SACM particularly suitable for real-time applications, where rapid processing is crucial. The significant reduction in runtime not only demonstrates the computational superiority of SC-based SACM but also its potential for large-scale deployment in practical scenarios.

Table 2. Computational time (s).

Piping Number	KM	GMM	FCM	SACM
1	6.9079	99.7843	227.2507	0.6001
2	6.21663	93.2404	214.3872	0.1907
3	5.6533	83.7551	217.9142	0.2122
4	6.6272	101.6670	203.7636	0.1717
5	6.3952	99.0716	201.9092	0.1611
6	7.3378	104.0337	207.2638	0.1708
7	6.5378	89.0131	204.2211	0.1601
8	8.4267	100.5110	197.8356	0.1638
Mean	6.7628	96.3845	209.3182	0.2288

4. Open Issues

4.1. Computing-Oriented Routing Protocol

In the (Com)²INet architecture, the essence of traffic scheduling is to coordinate and optimize communication and computation, forward computing tasks to the optimal network components, and achieve efficient use of multi-dimensional resources and end-to-end QoS and QoE guarantees. This requires comprehensive consideration of network status and available computing resource information to make optimal forwarding decisions to support traffic scheduling. In addition, how to effectively balance resources between communication and computation in routing decision-making to avoid resource waste or bottlenecks is still a complex optimization problem [31]. Especially in the case of multi-task concurrency, how to design a computing-oriented routing protocol is an important research topic to achieve ubiquitous connectivity and seamless collaboration of heterogeneous network technologies in (Com)²INet computing paradigm.

4.2. Deterministic Scheduling for Massive Computing Traffic

Services such as emergency communications and the metaverse require not only computing resources that are available at any time, but also deterministic guarantees for latency, packet loss, etc. to satisfy the strict QoS and QoE requirements. For example, in emergency communications scenarios, the real-time transmission of emergency disaster remote sensing and the rescue instructions requires that data transmission has strict delay and jitter constraints to ensure accurate feedback of disaster situation and timely release of instructions. The core challenge is how to achieve efficient, low-latency resource allocation and path selection for massive computing traffic in a dynamic, heterogeneous network environment [32]. Therefore, developing a deterministic scheduling mechanism for massive computing traffic to support advanced computing services effectively is an urgent issue.

4.3. Federated-Enabled Privacy Protection

Security and privacy are critical to all computing paradigms. Although cryptographic techniques such as differential privacy and homomorphic encryption have been used to improve system security and privacy, they may reduce the output accuracy and increase computational overhead. Information sharing and collaboration among multiple service providers make the (Com)²INet paradigm introduce security and privacy concerns. The core issue is how to achieve a balance between privacy protection and model performance in distributed data collaboration, and how to reduce overhead of computation and communication while ensuring privacy remains an open issue and deserves further study. As depicted in Figure 9, federated-enabled mechanism only shares model parameters instead of raw data [33], which is an important research topic for privacy protection and reducing data transmission costs.

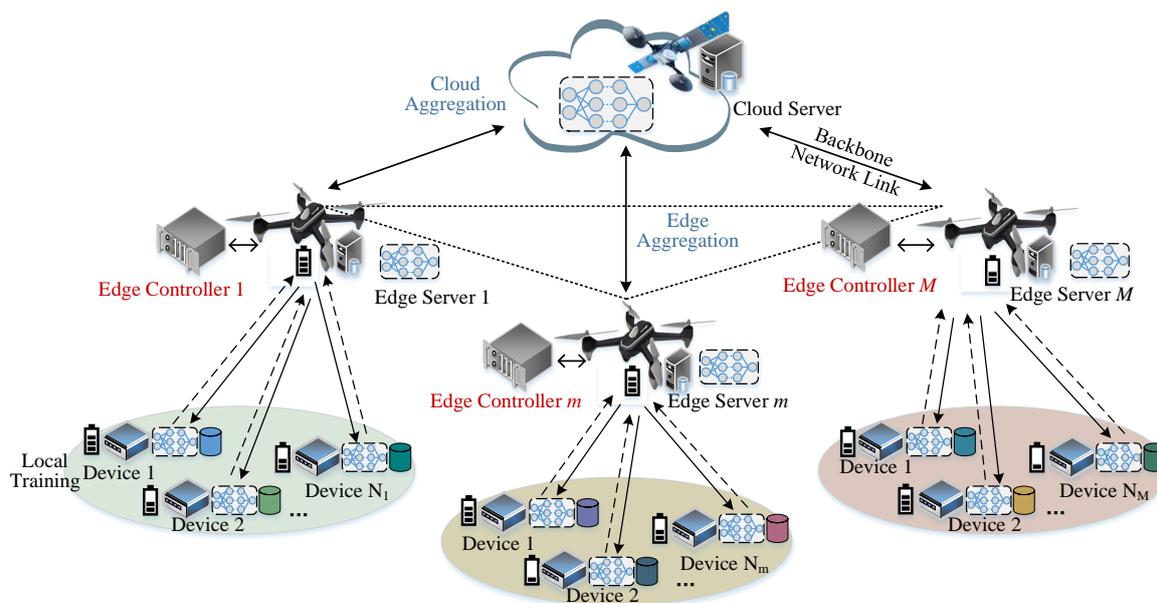


Figure 9. Illustration of federated-enabled privacy protection.

5. Conclusions

In this work, we have exhibited (Com)²INet architecture for effectively managing the mapping between multi-dimensional heterogeneous resources and the advanced computing services. Next, we have introduced the system architecture, key technologies and working principles of (Com)²INet. Furthermore, we have presented an use case of levee piping hazard inspection via remote sensing and proposed an SC-based SACM to make the optimal joint decision for levee piping hazard identification. Experimental results have confirmed the effectiveness of (Com)²INet architecture. Finally, we discuss the open issues and potential research trends. For the future work, we will study an adaptive levee piping hazard inspection mechanism for federated reinforcement learning to enable (Com)²INet to support data-sensitive computing services, as illustrated in Figure 9.

Author Contributions: Conceptualization, J.C. and L.G.; methodology, J.C.; software, L.G.; validation, J.C., L.G., Y. W. and K.W.; formal analysis, Y. W.; investigation, K.W.; resources, S.H.; data curation, Y. W.; writing—original draft preparation, J.C. and L.G.; writing—review and editing, J.C. and L.G.; visualization, H.S.; supervision, Z.W.; project administration, S.Y.; funding acquisition, S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the National Institute of Natural Hazards, Ministry Of Emergency Management of China under grant no.ZDJ2024-17, National Key R&D Program of China under grant no.2024YFC3013304, National Natural Science Foundation of China under grant no.U23B2026, no.61971458, no.52404225, Research grants from National Institute of Natural Hazards, Ministry of Emergency Management of China under grant no.ZDJ2024-29, Key Science and Technology Program Project of the Ministry of Emergency Management of China under grant no.2024EMST020201.

Data Availability Statement: The data presented in this study are temporarily unavailable due to confidentiality requirements related to levee emergency situations, and further research on the data will be conducted in the future.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

(Com) ² Inet	Communication and computation integrated network
SDN	Software defined network
NFV	Network function virtualization
QoS	Quality of service
QoE	Quality of experience
AI	Artificial intelligence
VNF	Virtual network function
SC	Service chain
UAV	Unmanned aerial vehicle
SACM	Stepwise adaptive clustering method
TIR	Thermal infrared
KM	k-Medoids
FCM	Fuzzy C-Means
GMM	Gaussian Mixture Model
mIoU	Mean Intersection over Union
NEDT	Noise Equivalent Temperature Difference

References

1. D. Yang; W. Zhang; Q. Ye; C. Zhang; N. Zhang; C. Huang. DetFed: Dynamic resource scheduling for deterministic federated learning over time-sensitive networks. *IEEE Transactions on Mobile Computing*, 2024, 23(5), 5162–5178.
2. Q. Tang; F. R. Yu; R. Xie; A. Boukerche; T. Huang; Y. Liu. Internet of intelligence: A survey on the enabling technologies, applications, and challenges. *IEEE Communications Surveys & Tutorials*, 2022, 24(3), 1394–1434.
3. S. Duan; D. Wang; J. Ren; F. Lyu; Y. Zhang; H. Wu. Distributed artificial intelligence empowered by end-edge-cloud computing: A survey. *IEEE Communications Surveys & Tutorials*, 2023, 25(1), 591–624.
4. J. Tang; M. M. Jalalzai; C. Feng; Z. Xiong; Y. Zhang. Latency-aware task scheduling in software-defined edge and cloud computing with erasure-coded storage systems. *IEEE Transactions on Cloud Computing*, 2023, 11(2), 1575–1590.
5. H. Djigal; J. Xu; L. Liu; Y. Zhang. Machine and deep learning for resource allocation in multi-access edge computing: A survey. *IEEE Communications Surveys & Tutorials*, 2022, 24(4), 2449–2494.
6. J. Liu; K. Xiong; D. W. K. Ng; P. Fan; Z. Zhong; K. B. Letaief. Max-min energy balance in wireless-powered hierarchical fog-cloud computing networks. *IEEE Transactions on Wireless Communications*, 2020, 19(11), 7064–7080.
7. W. Zhang; D. Yang; W. Wu; H. Peng; N. Zhang; H. Zhang. Optimizing federated learning in distributed industrial IoT: A multi-agent approach. *IEEE Journal on Selected Areas in Communications*, 2021, 39(12), 3688–3703.
8. Q. Ye; W. Shi; K. Qu; H. He; W. Zhuang; X. Shen. Joint RAN slicing and computation offloading for autonomous vehicular networks: A learning-assisted hierarchical approach. *IEEE Open Journal of Vehicular Technology*, 2021, 2, 272–288.
9. J. Chen; J. Chen; K. Guo; R. Hu; T. Zou; J. Zhu; H. Zhang. Fault tolerance oriented SFC optimization in SDN/NFV-enabled cloud environment based on deep reinforcement learning. *IEEE Transactions on Cloud Computing*, 2024, 12(1), 200–218.
10. Y. -W. Chen; C. -Y. Li; C. -C. Tseng; M. -Z. Hu. P4-TINS: P4-driven traffic isolation for network slicing with bandwidth guarantee and management. *IEEE Transactions on Network and Service Management*, 2022, 19(3), 3290–3303.
11. M. Xu; M. Jia; Q. Guo; T. de Cola. Delay-sensitive and resource-efficient VNF deployment in satellite-terrestrial networks. *IEEE Transactions on Vehicular Technology*, 2024, 73(10), 15467–15482.
12. K. Guo; J. Chen; P. Dong; S. Liu; D. Gao. FullSight: A feasible intelligent and collaborative framework for service function chains failure detection. *IEEE Transactions on Network and Service Management*, 2022, 19(4), 4546–4565.
13. Y. Zou; T. Pan; L. Lu; Z. Li; K. Yao; T. Huang. P4RSS: Load-aware intra-server load balancing with programmable switching ASICs. In *Proceedings of the IEEE International Conference on Communications (ICC)*, Rome, Italy, 28 May–01 June 2023; pp. 1893–1898.
14. M. Chiesa; R. Sedar; G. Antichi; M. Borokhovich; A. Kamisiński; G. Nikolaidis; S. Schmid. Fast reroute on programmable switches. *IEEE/ACM Transactions on Networking*, 2021, 29(2), 637–650.

15. R. Xie; Q. Tang; Q. Wang; X. Liu; F. R. Yu; T. Huang. Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues. *IEEE Network*, 2020, 34(3), 224–231.
16. W. Zhang; D. Yang; C. Zhang; Q. Ye; H. Zhang; X. Shen. (Com)²Net: A novel communication and computation integrated network architecture. *IEEE Network*, 2024, 38(2), 35–44.
17. L. Zeng; C. Yang; P. Huang; Z. Zhou; S. Yu; X. Chen. GNN at the edge: Cost-efficient graph neural network processing over distributed edge servers. *IEEE Journal on Selected Areas in Communications*, 2023, 41(3), 720–739.
18. Y. Shi; Y. Cao; J. Liu; N. Kato. A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks. *IEEE Network*, 2019, 33(1), 29–35.
19. Q. Qi; X. Chen; A. Khalili; C. Zhong; Z. Zhang; D. W. K. Ng. Integrating sensing, computing, and communication in 6G wireless networks: Design and optimization. *IEEE Transactions on Communications*, 2022, 70(9), 6212–6227.
20. M. C. Tsakiris; L. Peng; A. Conca; L. Kneip; Y. Shi; H. Choi. An algebraic-geometric approach for linear regression without correspondences. *IEEE Transactions on Information Theory*, 2020, 66(8), 5130–5144.
21. D. Li; S. Zhou; T. Zeng; R. H. Chan. Multi-prototypes convex merging based K-means clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 2024, 36(11) 6653–6666.
22. C. Qiao; M. Li; Y. Liu; Z. Tian. Transitioning from federated learning to quantum federated learning in internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2025, 27(1), 509–545.
23. C. Rong; J. H. Wang; J. Liu; J. Wang; F. Li; X. Huang. Scheduling massive camera streams to optimize large-scale live video analytics. *IEEE/ACM Transactions on Networking*, 2022, 30(2), 867–880.
24. S. Sundar; J. P. Champati; B. Liang. Multi-user task offloading to heterogeneous processors with communication delay and budget constraints. *IEEE Transactions on Cloud Computing*, 2022, 10(3), 1958–1974.
25. T. Mai; H. Yao; S. Guo; Y. Liu. In-network computing powered mobile edge: Toward high performance industrial IoT. *IEEE Network*, 2021, 35(1), 289–295.
26. H. Zhu; W. Jiang; Q. Hong; Z. Guo. When in-network computing meets distributed machine learning. *IEEE Network*, 2024, 38(5), 238–246.
27. Y. Wang; C. Yang; S. Lan; L. Zhu; Y. Zhang. End-Edge-Cloud collaborative computing for deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2024, 26(4), 2647–2683.
28. J. Chen; J. Chen; K. Guo. Queue-aware service orchestration and adaptive parallel traffic scheduling optimization in SDNFV-enabled cloud computing. *IEEE Transactions on Cloud Computing*, 2023, 11(4), 3525–3540.
29. M. Zhang, Y. Zhu, J. Liu, F. Wang, and F. Wang, CharmSeeker: Automated pipeline configuration for serverless video processing, *IEEE/ACM Transactions on Networking*, 2022, 306, 2730–2743.
30. X. Li, R. Xie, F. R. Yu, T. Huang, and Y. Liu, Advancing software-defined service-centric networking toward in-network intelligence, *IEEE Network*, 2021, 355, 210–218.
31. J. Zhang; A. Sinha; J. Llorca; A. M. Tulino; E. Modiano. Optimal control of distributed computing networks with mixed-cast traffic flows. *IEEE/ACM Transactions on Networking*, 2021, 29(4), 1760–1773.
32. D. Yang; K. Gong; J. Ren; W. Zhang; W. Wu; H. Zhang. TC-Flow: Chain flow scheduling for advanced industrial applications in time-sensitive networks. *IEEE Network*, 2022, 362, 16–24.
33. C. Qiao; M. Li; Y. Liu; Z. Tian. Transitioning from federated learning to quantum federated learning in internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2025, 27(1), 509–545.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.