Exploring LLM-Based Multi-Agent Situation Awareness for Zero-Trust Space-Air-Ground Integrated Network

Xinye Cao, *Graduate Student Member, IEEE*, Guoshun Nan, *Member, IEEE*, Hongcan Guo, Hanqing Mu, Long Wang, Yihan Lin, Qinchuan Zhou, Jiayi Li, Baohua Qin, Qimei Cui, *Senior Member, IEEE*, Xiaofeng Tao, *Senior Member, IEEE*, He Fang, *Member, IEEE*, Haitao Du, and Tony Q. S. Quek, *Fellow, IEEE*

***Abstract*—Space-air-ground integrated network (SAGIN), which integrates satellite systems, aerial networks, and terrestrial communications, offers ubiquitous coverage for a multitude of applications. Nevertheless, the highly dynamic and open nature of SAGIN increases the network's vulnerability. Hence, zero-trust security, operating on the principle of "never trust, always verify", holds the significant potential of securing SAGIN. However, implementing zero-trust SAGIN in practice presents three primary challenges: 1) understanding massive unstructured threat information across diverse domains, 2) performing adaptive security assessments, and 3) making in-depth security decisions. This motivates us to propose SAG-Attack and LLM-SA to enhance zero-trust SAGIN. SAG-Attack serves as a simulator that aims to mimic various attacks in SAGIN. Our LLM-SA is a novel situation awareness method that explores the multiple agents of large language model (LLM). Specifically, the output logs of SAG-Attack will be fed into LLM-SA, and LLM-SA fuses vast amounts of heterogeneous threat information from various domains, thus tackling the first challenge. Then, our LLM-SA relies on multiple LLM-based agents to perform adaptive security assessments, utilizing the chain-of-thought capabilities of LLMs to automatically generate in-depth defense strategies, thereby addressing the second and third challenges. Experiments on five benchmarks demonstrate the superiority of the proposed SAG-Attack and LLM-SA. Notably, our method based on open-sourced Llama3-8B even outperforms ChatGPT-4 under the same setting, despite involving significantly fewer parameters. To foster further research in this area, we will release our platform to the community, facilitating the advancement of zero-trust SAGIN.**

***Index Terms*—Space-air-ground integrated network, zero trust, LLM-based multi-agent, situation awareness.**

Received 2 June 2024; revised 26 February 2025; accepted 24 March 2025. Date of publication 11 April 2025; date of current version 30 May 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB2902200; in part by the National Natural Science Foundation of China under Grant 62471064; in part by the National Research Foundation, Singapore, and Infocomm Media Development Authority under its Future Communications Research and Development Program; and in part by Beijing Natural Science Foundation Program under Grant L232002. *(Hongcan Guo, Hanqing Mu, and Long Wang contributed equally to this work.) (Corresponding author: Guoshun Nan.)*

Xinye Cao, Guoshun Nan, Hongcan Guo, Hanqing Mu, Long Wang, Yihan Lin, Qinchuan Zhou, Jiayi Li, Baohua Qin, Qimei Cui, and Xiaofeng Tao are with the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: caoxinye@bupt.edu.cn; nanguo2021@bupt.edu.cn; ai.guohc@bupt.edu.cn; mhq@bupt.edu.cn; wl2023@bupt.edu.cn; cuiqimei@bupt.edu.cn; taoxf.bupt@gmail.com).

He Fang is with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350007, China (e-mail: fanghe@fjnu.edu.cn).

Haitao Du is with China Mobile Research Institute, Beijing 100084, China (e-mail: duhaitao@chinamobile.com).

Tony Q. S. Quek is with the Department of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore 487372, and also with the Yonsei Frontier Laboratory, Yonsei University, Seoul 03722, South Korea (e-mail: tonyquek@sutd.edu.sg).

Digital Object Identifier 10.1109/JSAC.2025.3560042

I. Introduction

A. Background

THE International Telecommunication Union (ITU) has recently published the framework of the sixth generation of communications, commonly known as 6G. Two new usage scenarios highlighted in the framework are massive communications and ubiquitous connectivity. Space-air-ground integrated network (SAGIN) [1], [2], which integrates satellite systems, aerial networks, and terrestrial networks, offers ubiquitous coverage for numerous applications. As a next-generation wireless networking paradigm, SAGIN has attracted increasing attention. Figure 1 illustrates the components of SAGIN: the space-based networks comprising geostationary Earth orbit (GEO) satellites, medium Earth orbit (MEO) satellites, and low Earth orbit (LEO) satellites, the air-based networks formed by multiple unmanned aerial vehicle (UAV) networks, and the ground-based networks including numerous wireless local area networks, cellular networks, and Ad Hoc networks.

While SAGIN's highly dynamic and open environment offers considerable benefits for global coverage, seamless communication, and emergency recovery, it also presents severe security issues, causing the network to be extremely fragile under various malicious attacks. Figure 1 illustrates that a malicious user can utilize jamming attacks [3] to interfere with wireless communications between LEO and ground station gateways. Unauthorized users can eavesdrop on confidential satellite information by intercepting wireless network traffic [4]. Other threats, including Distributed Denial-of-Service (DDoS) [5] and spoofing attacks [6], [7], can lead to service unavailability and data leakage of safe-critical applications such as smart health and internet of vehicles.

B. Motivation

The aforementioned vulnerability of SAGIN necessitates a redesign of the network security architecture against various

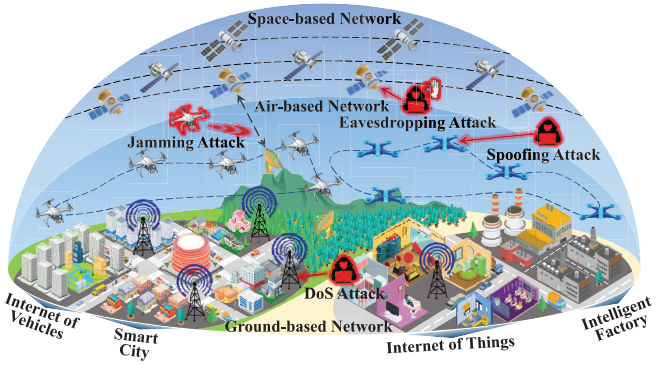


Fig. 1. Illustration of various attacks in space-air-ground integra jamming, eavesdropping, and spoofing attacks, where SAGIN integrates satellite systems, aerial networks, and terrestrial networks.

attacks under heterogeneous and dynamic environments. ITU also highlighted that security is one of the principles of next-generation networking [8]. The recently proliferated zero-trust paradigm [9], [10], [11], operating on the concept of "never trust, always verify", relies on continuous verification with context-aware, dynamic, and intelligent authentication schemes [12], [13], [14], thereby holding the promise of mitigating various risks in SAGIN. However, implementing zero-trust architecture within SAGIN presents three primary challenges:

1) **Understanding massive unstructured threat data:** Massive applications and devices in SAGIN, such as the Internet of Things (IoT) and mobile equipment, consistently generate huge amounts of context data, including system logs and threat information. It is challenging for a zero-trust SAGIN system to explore these massive unstructured data to understand the underlying interactions between diverse cues from various domains.
2) **Performing adaptive security assessments:** The assessment score [13], [15], [16] indicates the threat level of SAGIN and can facilitate access control decisions of zero-trust methods. It is challenging to obtain accurate scores due to the dynamics and heterogeneity of SAGIN with tens of thousands of devices involved in the network.
3) **Making in-depth security decisions:** There are massive threats in SAGIN that present complex interactions [17]. It is challenging for existing rule-based or deep learning-based defense strategies to capture such intricate relationships for making comprehensive and in-depth decisions [18], [19].

C. Our Method

The aforementioned issues motivate us to propose SAG-Attack and Large Language Model-based Situation Awareness (LLM-SA). LLM-SA is a novel situation awareness method that explores LLM-based agents to enhance zero-trust SAGIN. We outline six high-level design principles for SAG-Attack and LLM-SA to tackle the three challenges.

1) **Adaptive:** Compared to traditional perimeter-based situation awareness, the one for zero-trust SAGIN should be adaptive to fit the dynamic and open environments across space, air, and ground network segments.
2) **Learnable:** An LLM or deep-learning component is expected to be fine-tuned on unstructured threat data generated in SAGIN to meet customized security requirements.
3) **Collaborative:** As SAGIN works decentralized, the awareness modules for zero-trust SAGIN should be collaborative to make more comprehensive decisions.
4) **Pluggable:** Awareness components are expected to be pluggable for configuration, facilitating an administrator's replacement of modules for customized security settings.
5) **Large-scale:** As SAGIN involves thousands of devices and connections, the awareness method should be able to extract key features from massive threat information.
6) **Efficient:** The situation awareness method should be efficient enough to generate timely responses in large-scale SAGIN for massive threats across various domains.

Keeping the above goals in mind, we first design and implement SAG-Attack, a SAGIN simulator that mimics large-scale communications and threats across satellite networks, aerial networks, and terrestrial networks on the ns-3 [20] platform. The framework also involves a learnable neural component that can detect threat information in various domains, and this unstructured data will be fed into the proposed situation awareness method LLM-SA. Our LLM-SA first learns to extract salient attack features from information generated by SAG-Attack and then relies on multiple LLM-based agents to measure the threat level and generate defense strategies. For the adaptive and learnable goals, we can fine-tune LLMs for alignment with various SAGINs. These tunable LLMs are collaborative and pluggable and can be replaced to meet customized configurations in highly dynamic network environments, thus satisfying the second and third design principles. Finally, the output of LLM-SA will be fed into a zero-trust engine for authentication and security automation. Experiments on large-scale network simulation and real-world tests show the effectiveness of our proposed SAG-Attack and LLM-SA. Our code is publically available.[1]

D. Main Contributions

The main contributions of this paper are three-fold:

- **SAG-Attack Simulator:** We design and implement SAG-Attack, a novel simulator that can mimic various attacks in large-scale SAGIN. Our SAG-Attack consists of four components: satellite networks, UAV networks, ground-based networks, as well as a network monitor that collects threat information from various domains. We additionally develop the physical protocol DVB-S2X [21] for LEO satellites to facilitate the simulation of various attacks in satellite networks, as the implementation of the protocol is not publicly available in the ns-3 platform.
- **LLM-SA:** We present LLM-SA, a novel situation awareness method that relies on LLM-based agents for

[1]https://github.com/caoxinye/LLM-SA\#

zero-trust SAGIN. Specifically, the proposed LLM-SA summarizes the massive threat logs with an LLM, then proceeds to build correlations between attacks and extract key features with a security-weighted Principal Component Analysis (SW-PCA) algorithm, thus properly tracking the first challenge. Multiple LLM-based agents are employed to address the second and third challenges raised in Section I-B. We combine our agents with a mathematical method to ensure the efficiency of the LLM-SA. Our SAG-Attack and LLM-SA also meet the six criteria discussed in Section I-C. To the best of our knowledge, we are the first to explore LLM-based agents to enhance situation awareness of zero-trust SAGIN.

- **Experiments:** We conduct extensive experiments on four public benchmarks to show the effectiveness of the proposed SAG-Attack simulator and LLM-SA, yielding the state-of-the-art defense for zero-trust SAGIN. We also build a more comprehensive dataset on the SAG-Attack simulator, and such a dataset can serve as a benchmark for zero-trust SAGIN. Furthermore, we conduct three types of cyberattacks on real-world tests to demonstrate the practical potential of LLM-SA. Finally, we provide a case study to visually demonstrate the detailed work procedure of the proposed LLM-SA and give some insightful discussions based on our observations.

E. Related Work

1) Zero Trust in SAGIN: Zero-trust architecture has emerged as a pivotal paradigm in the field of cybersecurity [9], [10], [11]. Numerous studies have investigated multi-factor continuous authentication for zero trust [12], [13], [14]. With regard to zero trust in satellite networks, Fu et al. devised a continuous authentication mechanism for satellite networks [22]. Toward zero trust in air-based networks, architectures [23], authentication schemes [24], and trust monitoring [25] have been investigated in previous works. To the best of our knowledge, we are the first to explore large language model-based multiple agents for zero-trust SAGIN.

2) Network Security Situation: The main technologies used in the network security situation assessment method include mathematical statistics, knowledge reasoning [15], [16], [26], and pattern recognition [27], [28]. As for situation awareness in communication networks, Klement et al. enabled the MITRE ATT&CK framework to assess threats in 6G Radio Access Networks [29]. As for zero-trust situation awareness, Chen et al. proposed a security awareness and protection system that leverages zero-trust architecture for a 5G-based smart medical platform, with the environment regarded as a key dimension [18]. Dai et al. proposed a mobile Internet network security situational awareness model that incorporates privacy differentiation analysis and user entity behavioral analytics [19]. Seaton et al. introduced path-aware risk scores for access control, which accounts for risks along the network path that requests traverse from source to destination [30]. Two key differences between our work and the previous ones are: 1) we develop a simulator to mimic various attacks in SAGIN, and 2) we propose an LLM-based situation awareness method for zero-trust SAGIN.

II. Our SAG-Attack Simulator

To simulate highly dynamic and heterogeneous networks of SAGIN, we develop a SAG-Attack simulator based on ns-3. The proposed SAG-Attack simulator can generate malicious traffic to mimic the threats in real-world scenarios. Our platform consists of four components: satellite systems, aerial networks, terrestrial networks, and a network monitor comprising multiple detection modules. The key ingredient is the construction of attack scenarios of satellites and UAVs. We also integrate a zero-trust platform [31] into our SAG-Attack simulator. We detail each module as follows.

A. Satellite Systems

We build different types of satellites, including a GEO satellite and several LEO satellites, to comprehensively simulate various satellite communication scenarios. We have developed the physical layer protocol according to DVB-S2X. For various communication tasks, our simulation parameters support dynamic configuration, including frequency, bandwidth, modulation mode, etc. Additionally, we have configured the network and application layers to mimic data transmission between satellites, as well as between satellites and ground nodes, facilitating the simulation of diverse application scenarios.

B. Aerial Networks

We deploy multiple UAV nodes and equip them with WLAN protocols to improve the communication capability of ground users. In addition, we configure the MobilityModel in ns-3 to enable a random walk mobility pattern for these UAVs, which increases the dynamic nature of the network and better accommodates different types of network conditions.

C. Terrestrial Networks

The terrestrial networks module includes the ground base stations (eNodeBs) and the user equipment nodes (UEs). Among them, eNodeBs are distributed in a mesh topology, with UEs located around satellites, UAVs, and eNodeBs. We install the LTE protocol stack for UEs and eNodeBs to enable communication between them. We can seamlessly integrate the simulated attack application with this module to launch attacks on nodes at all layers to simulate real attack traffic and log messages.

D. Attack Scenarios Simulation

We create a malicious traffic generation application for each attack and install it on the designated attack node to simulate attack scenarios in SAGIN. For DDoS attacks, we create an application that will send a large amount of traffic and install it on 100 different attack nodes to launch a joint attack on ground eNodeBs, UAVs, or satellite nodes, to truly reproduce the cross-layer DDoS attack behavior. Similarly, for DoS attacks, we install the above application on an attack node to simulate DoS attacks. For infiltration, brute force, spoofing, and recon, we also build corresponding malicious traffic generation applications and install them on specific attack

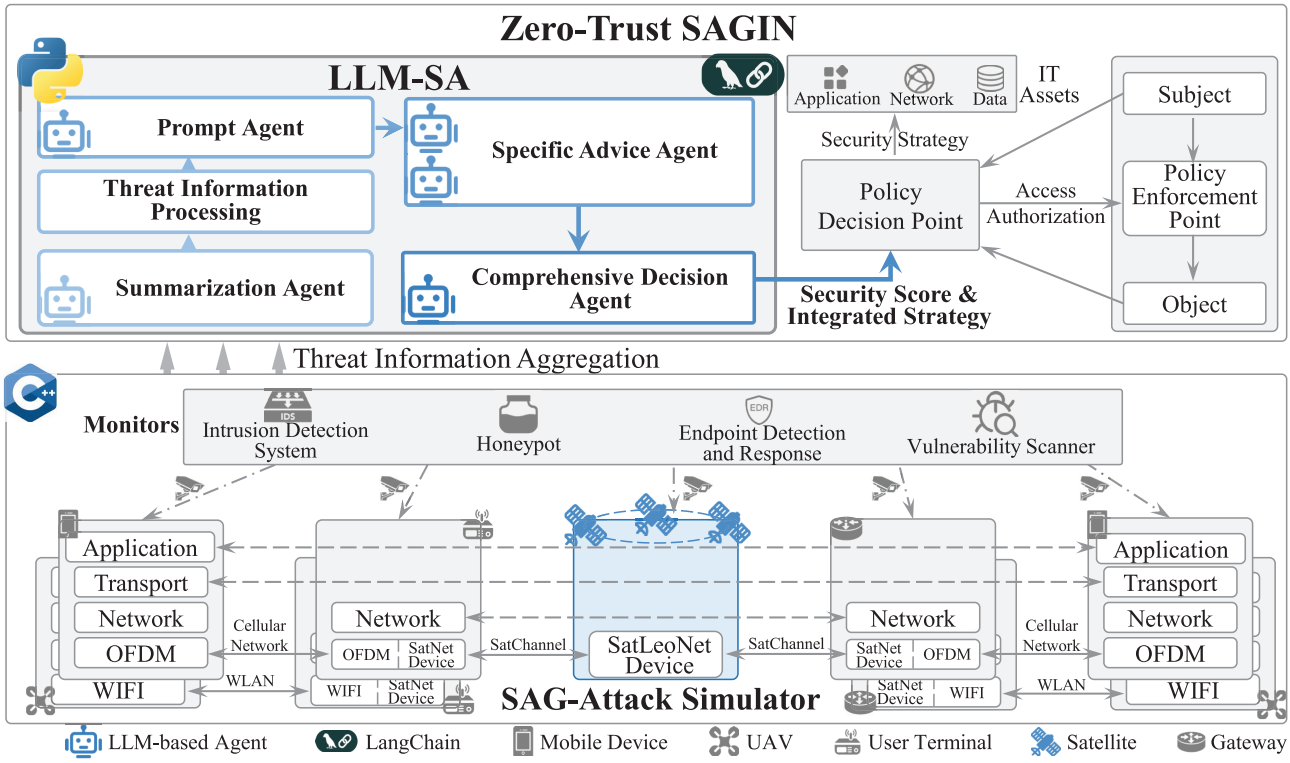


Fig. 2. Illustration of situation awareness in zero-trust access control architecture. The Situation Awareness Module aggregates threat information and processes it through threat information processing. It ultimately generates a security score and integrated strategy provided to the Policy Decision Point (PDP). PDP comprehensively considers the characteristics of objects and subjects, as well as the security score and integrated strategy, ultimately generating a security strategy and access authorization. These are then provided to IT assets and the Policy Enforcement Point (PEP).

nodes to simulate these attack behaviors. For the web attack, we construct numerous HTTP request packets representing various web attack vectors. These packets are then used in an application we built to simulate the attacker's actions.

III. System Model

As illustrated in Figure 2, the proposed zero-trust SAGIN is divided into three parts: the SAG-Attack simulator, the LLM-SA, and the zero-trust access control module. Building upon the SAG-Attack simulator, the system model section provides a comprehensive overview of the components of the LLM-SA model and its application within a zero-trust SAGIN architecture. The SAG-Attack Simulator is capable of simulating various types of attacks in SAGIN. The collected threat information is fed into the Zero-Trust SAGIN framework. LLM-SA then classifies, correlates, and evaluates the overall network environment, and provides corresponding security strategies. The security scores and strategies output by LLM-SA are subsequently fed into the Policy Decision Point of the zero-trust architecture, facilitating more comprehensive security decision-making. This model enables the system to efficiently process and respond to threat information across satellite, aerial, and ground networks, thereby enhancing overall network security through improved situational awareness and real-time defense capabilities.

A. Zero-Trust Access Control Model

We develop a SAG-Attack simulator based on the ns-3 platform, simulating various attack scenarios to provide data support for the zero-trust access control model. This model is based on the zero-trust architecture proposed by the National Institute of Standards and Technology (NIST) [31]. Unlike previous works, this paper focuses on the network security situational awareness module. This module collects information from the network environment, ultimately generating security levels and security policies against current network attacks, which are then provided to the policy decision point. The policy decision point, based on the principle of least privilege, comprehensively considers the contextual information of the access subjects, the importance of the access objects, and the security levels and security policies provided by the LLM-SA. Security levels dynamically adjust access control policies, and security strategies can respond to potential security threats. The access control policies generated by the policy decision point are distributed to the policy enforcement point, where they regulate the access of subjects to objects, enhancing the security and integrity of IT assets.

B. LLM-Based Hierarchical Collaborative Architecture for Situation Awareness

As shown in Figure 3, we propose an LLM-based hierarchical collaborative architecture for situation awareness in SAGIN. The input is attack information collected from various detection models. The output is the security level of the environment and security policies. To address the challenges of cross-domain threat analysis in SAGIN, our design integrates attack information from satellite, aerial, and ground networks, enabling the identification of inter-network attack

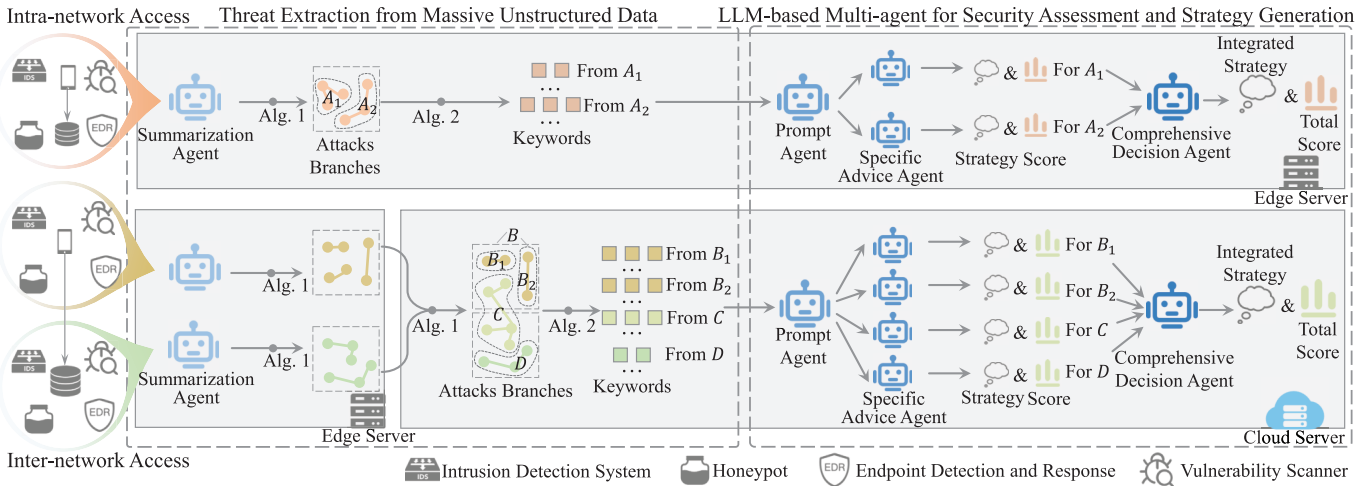


Fig. 3. Illustration of LLM-SA. The upper and bottom parts of this figure, respectively, describe the scenarios of intra-network access and inter-network access. The whole process is divided into threat information processing and LLM-based Multi-agent for security assessment and strategy generation. (Alg. 1: S&F attack correlation; Alg. 2: SW-PCA for Key Information Extraction; A_1, A_2, B_1, B_2, C, D corresponding to each set of correlated attacks).

patterns and enhancing the context awareness of security policies. Considering the distributed sub-networks in SAGIN, we define the access within a sub-network as intra-network access and the access across sub-networks as inter-network access. Specifically, the process can be mainly divided into four steps.

1) Threat Information Processing: Detection systems, including intrusion detection systems (IDS),[2] gateways, vulnerability scanners, endpoint detection and response (EDR), and honeypots, are deployed within every sub-network in a distributed manner to realize real-time monitoring of current network conditions. The LLM-based Summarization Agents extract this multisource threat information into explicit descriptions of attacks and their corresponding key characteristics. Assume that there are N distributed sub-networks in SAGIN, and J_i denotes the total number of threat information pieces within the i-th sub-network N_i: $D_i = \{d_j^i \mid j \in [1, J_i]\}, i \in [1, N]$, where D_i refers to the set of threat information in N_i, and d_j^i represents the j-th piece of threat information in D_i.

The S&F attack correlation algorithm f_a further associates and clusters the threat information in N_i to generate an attack correlation graph. The set of connected components $\mathcal{G}_i$ within this graph can be represented as:

$$\begin{aligned}\mathcal{G}_i &= f_a\left(D_i \mid w_i^a\right) \\ &= \left\{g_k^i \mid k \in [1, K_i]\right\}, i \in [1, N], \end{aligned} \tag{1}$$

where w_i^a is a parameter in f_a that controls the relative importance of semantic information compared to key characteristics in the correlation decision. g_k^i represents the k-th connected component in $\mathcal{G}_i$, consisting of elements d_j^i from the set D_i that are similar in both semantics and key features.

Specifically, for inter-network access, we merge elements from the two sub-network attack correlation graphs and apply the algorithm f_a again to generate a new attack correlation graph representing the inter-network attack correlations. For inter-network access between sub-networks D_{i_1} and D_{i_2}, the union of their threat information D_{i_1,i_2} and the set of connected components $\mathcal{G}_{i_1,i_2}$ in the resulting attack correlation graph can be represented as: $D_{i_1,i_2} = D_{i_1} \cup D_{i_2}$, $\mathcal{G}_{i_1,i_2} = f_a(D_{i_1,i_2} \mid w_{i_1,i_2}^a)$ where w_{i_1,i_2}^a is a parameter in f_a for inter-network access.

To extract key information and reduce the dimensionality of the data, we apply a Principal Component Analysis (PCA)-based algorithm f_b to filter and refine the data. The processed data can be represented as $\mathcal{L}_i$ and $\mathcal{L}_{i_1,i_2}$, which correspond to intra-network access and inter-network access scenarios, respectively:

$$\begin{aligned}\mathcal{L}_i &= f_b\left(\mathcal{G}_i \mid w_i^b\right) \\ &= \left\{l_k^i \mid k \in [1, K_i]\right\}, i \in [1, N], \end{aligned} \tag{2}$$

$$\mathcal{L}_{i_1,i_2} = f_b\left(\mathcal{G}_{i_1,i_2} \mid w_{i_1,i_2}^b\right), \tag{3}$$

where w_i^b and w_{i_1,i_2}^b are parameters in f_b that control the information extraction process.

2) LLM-Based Multi-Agent for Security Assessment and Strategy Generation: A prompt [32] is a piece of text or instruction given to a language model to guide its response or to generate a specific type of output. The Prompt Agent accesses the information of each connected component g_k^i, and subsequently generates a prompt that describes the problem to be solved with this set of related threat information. We abstract the Prompt Agent as a function f_c, whose output consists of a prompt p_k^i for the problem description and a summary c_k^i of the threat description: $\{p_k^i, c_k^i\} = f_c(l_k^i)$. Denote P_i and C_i as the collections of prompts and threat descriptions, respectively:

$$P_i = \left\{p_k^i \mid k \in [1, K_i]\right\}, i \in [1, N], \tag{4}$$

$$C_i = \left\{c_k^i \mid k \in [1, K_i]\right\}, i \in [1, N]. \tag{5}$$

3) Security Assessment and Strategy: We design a dynamic LLM-based multi-agent system for security assessment and

[2]Suricata: https://github.com/OISF/suricata

strategy generation. For each pair $\{p_k^i, c_k^i\}$ and the corresponding expert knowledge in the vector database, an LLM-based intelligent agent a_k^i is generated to evaluate the threat level of associated attacks within the connected component, resulting in a security level s_k^i and strategy t_k^i.

Vector databases store text data and their vector representations (e.g., word embeddings) in a high-dimensional space. Based on the similarity between vectors and efficient index structures (such as trees or hash tables), vector databases can quickly retrieve texts that are highly similar to the target text. We incorporate the Common Vulnerability Scoring System (CVSS) [29] in the vector database. When constructing prompts, we retrieve relevant texts in the vector database and feed them into the LLM. CVSS decomposes network security assessment into multiple metrics, covering almost all aspects of the network security evaluation, thereby enhancing the systematic and comprehensive nature of LLM-based security assessments. More details of the CVSS metric are given in Section IV-A and Appendix A.

The agent is dynamically generated according to each set of related network attacks currently identified. As the network environment changes, the agent will also adjust accordingly to adapt massive threat information in SAGIN. The collection of security levels and strategies can be represented as vectors $\boldsymbol{S_i}$, and $\boldsymbol{T_i}$ respectively: $\boldsymbol{S_i} = \left[s_1^i\ s_2^i \cdots s_{K_i}^i\right]$, $\boldsymbol{T_i} = \left[t_1^i\ t_2^i \cdots t_{K_i}^i\right]$.

Given that each agent generates a security level and strategy from a distinct perspective, we design a more objective Comprehensive Decision Agent, which possesses comprehensive global information and determines the corresponding weights $\boldsymbol{W_i}$ for each agent by examining their identities: $\boldsymbol{W_i} = \left[w_1^i\ w_2^i \cdots w_{K_i}^i\right]$. The Comprehensive Decision Agent integrates the strategies $\boldsymbol{T_i}$ from each agent a_k^i and combines the prior knowledge contained in the vector database to generate the final security strategy. The final network security level SL_i is calculated as the weighted sum of the scores from each agent a_k^i, which can be represented as:

$$SL_i = \boldsymbol{S_i}^T \boldsymbol{W_i} = \sum_{k=1}^{K_i} s_k^i w_k^i. \tag{6}$$

IV. Our Proposed LLM-SA

We first explain preliminary knowledge of CVSS and then introduce three algorithms to enhance situation awareness, including an attack correlation algorithm based on semantic and feature similarity, a key information extraction algorithm utilizing PCA and frequency analysis, and a dynamic LLM-based multi-agent algorithm for situation assessment and security strategy generation.

A. Preliminary

We apply the CVSS [29] v3.1 standard as an component within the LLM-SA to enhance the systematic and comprehensive nature of the LLM-SA, which will be used in section IV-D. The CVSS is comprised of three metric groups: Base, Temporal, and Environmental. We use the Base metric for scoring. Base metrics represent the intrinsic qualities of a vulnerability that are constant over time and across user environments. This group is further divided into two subgroups: Exploitability and Impact.

TABLE I
CVSS Ratings

Severity	Score Range
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Medium	9.0 - 10.0

Exploitability metrics include Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI), and Scope (S). Impact metrics include Confidentiality (C), Integrity (I), and Availability (A). Most of the test scopes remain unchanged, so we have set the scope vector to "Unchanged" as the default setting. Detailed description of metric and numerical values of each metric are given in Appendix A. ISCBase in Formula 7 denotes Impact Sub Score. The CVSS score is calculated using the following formula for the Base Score:

$$\text{ISCBase} = 1 - [(1 - \text{C}) \times (1 - \text{I}) \times (1 - \text{A})], \tag{7}$$

$$\text{Impact} = 6.42 \times \text{ISCBase}, \tag{8}$$

$$\text{Exploitability} = 8.22 \times \text{AV} \times \text{AC} \times \text{PR} \times \text{UI}, \tag{9}$$

$$\text{BaseScore} = \lceil (\min(\text{Impact} + \text{Exploitability}, 10)) \rceil \tag{10}$$

Our proposed LLM-SA evaluates each vector within its scope and assigns a metric value, such as [AV, AC, PR, UI, Conf, Integ, Avail]. Subsequently, we utilize the scoring formula outlined in the CVSS standard to compute the BaseScore. We can convert scores into severity ratings through Table I.

B. Attack Correlation

Algorithm 1 is designed to partition the threat information dataset D_i into clusters based on semantic and feature similarities, utilizing a threshold θ_s to determine the grouping. Algorithm 1 categorizes the threat information into an attack correlation graph, which assists subsequent algorithms in uncovering potential attack correlations.

1. **Initialization of Groups:** Initially, each piece of threat information in the dataset D_i is treated as a distinct group: $\mathcal{G}_i = \left\{\left\{d_j^i\right\} \mid j = 1, 2, \ldots, J_i\right\}$, where d_j^i represents an individual piece of threat information within D_i.
2. **Computation of Similarities:** For each pair of threat information (d_{j1}^i, d_{j2}^i) in D_i, the Algorithm 1 computes both semantic and feature-based similarities:
   The natural language processing (NLP) model for semantic similarity SMT_CALC is employed to calculate a semantic similarity b_s which reflects the underlying connection of the threat information:

$$b_s = \text{SMT\_CALC}\left(d_{j1}^i, d_{j2}^i\right). \tag{11}$$

The Extract function Extract($[.*], d_j^i$) uses regex patterns to extract key features $(\beta^{(1)}, \beta^{(2)})$ from each pair (d_{j1}^i, d_{j2}^i):

$$\left(\beta^{(1)}, \beta^{(2)}\right) = \left(\text{Extract}\left([.*], d_{j1}^i\right), \text{Extract}\left([.*], d_{j2}^i\right)\right), \tag{12}$$

$$\left(|\beta^{(1)}|, |\beta^{(2)}|\right) = (m, n). \tag{13}$$

The function FTR_CALC, measuring the similarity between strings, is computed between each feature pair to create a similarity matrix S: $S_{i,j} = \text{FTR\_CALC}(\beta_i^{(1)}, \beta_j^{(2)})$. Each element of S quantifies the dissimilarity between corresponding features, forming a comprehensive matrix of pairwise feature similarity.

3. **Weighted Average of Similarity Matrix:** The feature-based similarity, f_s, is derived by calculating the weighted average of the similarity matrix S:

$$f_s = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} S_{i,j}^2}{\sum_{i=1}^{m}\sum_{j=1}^{n} S_{i,j}}. \tag{14}$$

We use $S_{i,j}$ itself as a weight to amplify the effect of high correlation values. Consequently, even a few similar keywords in the two pieces of information will be sensitively captured, and the correlation score will be significantly enhanced.

Algorithm 1 S&F Attack Correlation

Input: Dataset D_i, True labels Y_i, Similarity threshold θ_s
Output: partitioned dataset $\mathcal{G}_i$

```
G_i ← {d_j^i}
for each pair (d_j1^i, d_j2^i) in D_i do
    b_s ← SMT_CALC(d_j1^i, d_j2^i)
    (β^(1), β^(2)) ← (Extract([.*], d_j1^i), Extract([.*], d_j2^i))
    for each β_i^(1) in β^(1) do
        for each β_j^(2) in β^(2) do
            S_{i,j} ← FTR_CALC(β_i^(1), β_j^(2))
        end for
    end for
    f_s ← WeightedAverage(S)
    single_similarity ← (a×b_s + b×f_s)/(a+b)
    if single_similarity ≥ θ_s then
        δ_j1^i ← δ_j1^i ∪ δ_j2^i
    end if
end for
G_i ← {δ_1^i, δ_2^i, ..., δ_{K_i}^i}
return: G_i
```

4. **Combined Similarity Score and Group Merging:** To integrate both semantic and feature similarities, a combined similarity score $single\_similarity$ is computed as a weighted sum:

$$single\_similarity = \frac{a \times b_s + b \times f_s}{a + b}, \tag{15}$$

where $a = w_i^a$ and $b = 1 - w_i^a$ are coefficients that balance the contributions of semantic and feature similarities, respectively. If $single\_similarity$ meets or exceeds the threshold θ_s, the groups containing d_{j1}^i and d_{j2}^i, denoted as δ_{j1} and δ_{j2}, respectively, are merged into δ_{j1}: $\delta_{j1}^i \leftarrow \delta_{j1}^i \cup \delta_{j2}^i$. After iterating through all threat information pairs, the resulting set of groups is obtained:

$$\mathcal{G}_i = \left\{g_k^i \mid k \in [1, K_i]\right\} = \left\{\delta_1^i, \delta_2^i, \ldots, \delta_{K_i}^i\right\}. \tag{16}$$

Algorithm 2 SW-PCA Key Information Extraction

Input: Data branch set $\mathcal{G}_i$, frequency threshold w_i^b
Output: Set of key information $\mathcal{L}_i$

```
for each branch g_k^i in G_i do
    for each threat information d_{t,k}^i in g_k^i do
        s_t ← Extract([.*], d_{t,k}^i)
        S_k^i ← S_k^i ∪ {s_t}
    end for
    M ← TF-IDF(S_k^i)
    γ_k ← SW-PCA(M)
    F_{S_k^i} ← Frequency(S_k^i)
    η_k ← TopC_Percent(F_{S_k^i}, w_i^b)
    for each string s_k in S_k^i do
        if δ(s_k, γ_k) > τ then
            L_i ← L_i ∪ {(s_k, η_k)}
        end if
    end for
end for
return: L_i
```

C. Key Information Extraction

Algorithm 2 employs SW-PCA, which assigns weights based on sensitive words contained in key information, ultimately extracting the most critical components from massive threat information in $\mathcal{G}_i$.

1. **Extraction of Key Information:** Each threat information $d_{t,k}^i$ within a subgraph g_k^i is processed to extract key information based on a predefined regex pattern. The $\mathcal{S}_i$ can be denoted as:

$$\mathcal{S}_i = \bigcup_{k=1}^{K_i}\bigcup_{t=1}^{n} \text{Extract}\left([.*], d_{t,k}^i\right). \tag{17}$$

Here, n represents the number of threat information within the subgraph g_k^i. The extracted key information is added into the set $\mathcal{S}^i$.

2. **Conversion to TF-IDF Matrix:** The set $\mathcal{S}_i$ is converted into a term-frequency inverse document frequency (TF-IDF) matrix, M. The TF-IDF value is calculated for each key information in each threat information, providing a weight that signifies the importance of the key information in the threat information relative to its commonality across all threat information in the subgraph g_k^i:

$$M[m, n] = \text{TF-IDF}\left(key_n, threat_m, \mathcal{S}_i\right), \tag{18}$$

where m indexes the key information, n indexes the key information set of each threat information across

the set $\mathcal{S}_i$ and TF-IDF is a function that measures the importance of the key information in the key information set of threat information.

3. **SW-PCA:** Multiply the elements corresponding to matrix *M* by the elements corresponding to the weight matrix *SW* to obtain the weighted matrix M', then use the PCA algorithm on matrix M' to select the most essential key information.
   The weighted matrix M' is obtained by multiplying each element of the TF-IDF matrix *M* by the corresponding element in the weight matrix *SW*:
   $$M'[m,n] = M[m,n] \cdot SW[m,n], \tag{19}$$
   where $M[m,n]$ is the TF-IDF value for key information *m* and key information set of threat information *n*, and $SW[m,n]$ is the product of the weights corresponding to all sensitive information contained in the key information *m*:
   $$SW[m,n] = sw_m^1 \cdot sw_m^2 \cdot \ldots, \tag{20}$$
   where sw_m^i is the weight corresponding to the i-th sensitive information contained in the key information *m*. And now, the generated matrix M' contains weighted TF-IDF values, which integrate the frequency of each key information in the overall threat information and the security importance of each key information.
   PCA is applied to M' to reduce dimensionality by selecting the p principal components with the largest eigenvalues, as represented by $\gamma_k = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p\}$, which capture the most significant variance.
4. **Frequency Analysis and Key Information Aggregation:** The frequency of each string in $\mathcal{S}_k^i$ is calculated:
   $$F_{\mathcal{S}_k^i} = \text{Frequency}\left(\mathcal{S}_k^i\right). \tag{21}$$
   The top $w_i^b\%$ of frequently occurring terms, η_k, are identified:
   $$\eta_k = \text{TopC\_Percent}\left(F_{\mathcal{S}_k^i}, w_i^b\right). \tag{22}$$
   This key information is considered to be the most frequently occurring information globally. The global significance of the strings is evaluated based on their alignment with the principal components, quantified by the projection metric:
   $$\delta(s_k, \gamma_k) = \sum_{j=1}^{p} |\mathbf{v}_j^T s_k|. \tag{23}$$
   The key information s_k that exceed a predefined threshold τ in terms of their δ value are added to the key information set $\mathcal{L}_i = \mathcal{L}_i \cup \{(s_k, \eta_k) \mid \delta(s_k, \gamma_k) > \tau\}$.

D. Dynamic LLM-Based Multi-Agent for Situation Awareness

In the provided Algorithm 3, we compute the environmental safety score *c* by leveraging key data extracted from a specified pool. This procedure involves a chain-of-thought process with an artificial agent that processes prompts from this data, enabling a thorough and nuanced assessment. Subsequently, a score is assigned using established tools and knowledge. This chain-of-thought method systematically guides the model's reasoning process, thereby enhancing its interpretability and accuracy in problem-solving.

Algorithm 3 Dynamic LLM-Based Multi-Agent for Situation Awareness

Input: Set of tuples $\mathcal{L}_i$ containing high-frequency key information, Data pool $\mathcal{G}_i$
Output: Environmental safety score c

```
Initialize c_numerator ← 0, c_denominator ← 0
Initialize P_i ← ∅, S_i ← ∅
for each (l_k^i, g_k^i) in L_i × G_i do
    p_k^i, d_k^i ← Prompt_Agent(l_k^i, g_k^i)
    P_i.add(p_k^i), S_i.add(d_k^i)
    Agent_i ← Prompt_Agent(p_k^i)
    weight_i ← Specific_Advice_Agent(Agent_i, P_i)
    c_i ← Agent_i(S_i)
    c_numerator ← c_numerator + (weight_i × c_i)
    c_denominator ← c_denominator + weight_i
end for
c ← c_numerator / c_denominator
return: c
```

1. **Initialization and Data Preparation:** Key information strings, denoted as $\mathcal{L}_i$, are processed to generate prompts and associated threat information from the data pool $\mathcal{G}_i$. This step involves an interactive function Prompt_Agent, which translates key information into a usable format for further analysis: $(p_k^i, d_k^i) = \text{Prompt\_Agent}(l_k^i, g_k^i)$, where *p* represents the prompt derived from l_j^i and g_j^i, and d is the associated description. Sets *P* and *S* store the prompts and data respectively:
   $$P_i = \bigcup_k \{p_k^i\}, \quad S_i = \bigcup_k \{d_k^i\}. \tag{24}$$
2. **Agent Prompt Processing:** Each prompt p_i from the set P_i is inputted to a function Prompt_Agent which instantiates an agent $Agent_i$ for each prompt: $Agent_i = \text{Prompt\_Agent}(p_i)$. These agents are collected into a group $Agent\_group$:
   $$Agent\_group = \bigcup_p \{Agent_i\}. \tag{25}$$
3. **Scoring by Agents:** Each agent $Agent_i$ is evaluated based on its ability to interpret and respond to its corresponding prompt: $weight_i = \text{Specific\_Advice\_Agent}(Agent_i, P_i)$, where the Comprehensive Decision Agent will give a weight $weight_i$ based on the logical relationship between the prompts of each $Agent_i$ and their respective importance.
4. **Computation of Agent-Specific Scores:** After scoring, each agent will use its corresponding importance score $weight_i$ to weight the ratings based on the description set *S*. The most important thing is that $Agent_i$ can use the vector database and the CVSS calculator

TABLE II
PARAMETERS

Category	Parameter	Value
SAG-Attack	Altitude of LEO satellites	600 km
	Speed of LEO satellites	7,560 m/s
	Altitude of GEO satellites	35,786 km
	Altitude of UAVs	100 m
	Num of satellites	11
	Num of UAVs	100
	Num of eNodeBs	200
	Num of UEs	10,000
Attack correlation	Num of test instances	3,000
LLM-SA	Num of agent train instances	1,000
	Num of test instances	200

to give professional, unified, and reasonable scores: $c_i = Agent_i(S_i)$. The function $Agent_i(S_i)$ quantifies the environmental security by the agent $Agent_i$ based on processed data through a series of tools.

5. **Aggregate Environmental Safety Score Calculation:** The final Environmental Safety Score c is computed as a weighted average, where

$$c = \frac{\sum_i (weight_i \times c_i)}{\sum_i weight_i}. \tag{26}$$

This formulation ensures that agents with superior performance exert proportionally more influence on the score, indicative of their enhanced reliability and accuracy in assessment.

The chain-of-thought method, guided by our carefully designed prompts, maximizes the utilization of available data under controlled and interpretable conditions. Additionally, it dynamically adapts to the quality of agent responses, ensuring the validity and accuracy of environmental safety assessments.

To obtain continuous security assessments, we combine LLM-SA with the mathematical method NABC [33]. Specifically, the large number of parameters in the LLMs leads to longer time intervals between score updates. To solve this issue, we input the scores generated by the LLM-SA into the NABC method as initial values. The NABC method then performs continuous security assessments during the intervals between updates of the LLM-SA. Compared to using the NABC method alone, the integration of LLM-SA and NABC improves the accuracy of the security scoring while maintaining almost real-time performance. The detailed process of NABC is shown in Appendix B.

V. Experiments and Numerical Results

A. Experimental Setup

We run our network model on a server with the Ubuntu 20.04.5 operating system, equipped with 64GB of memory, an Intel(R) Xeon(R) Silver 4210 CPU, and two NVIDIA A40 GPUs. The simulation of attack scenarios is conducted using ns-3 v3.37. We also deploy and fine-tune the Llama3-8B[3] and Llama3-70B models on a server with the Ubuntu 20.04.5 operating system, equipped with 64GB of memory, an Intel(R) Xeon(R) Silver 4210 CPU, and two NVIDIA A800 GPUs. Instruction tuning for the LLMs is conducted using the LoRA (Low-Rank Adaptation) method [34]. Table II lists the specific parameters of the simulation for our experiments on SAGIN.

For the LoRA fine-tuning, we set the rank r to be 8, the scaling factor α to be 32, and the dropout rate to be 0.1. All other parameters for LoRA are set to their default values as provided by the `LoraConfig` function in the Huggingface PEFT library. For the fine-tuning process, we set the learning rate to 1e-4 and the number of training epochs to 10. The remaining training parameters are kept at their default values as specified by the `TrainingArguments` function.

To simulate the massive communication scenario within SAGINs, we set up one GEO satellite node, 10 LEO satellite nodes, 100 UAVs, 200 ground eNodeBs, and 10,000 UEs for the sub-network. UEs, UAVs, and satellites are respectively equipped with LTE protocol, WLAN protocol, and DVB-S2X protocol respectively. UAV nodes and LEO satellite nodes are configured with MobilityModel. The height of the UAV, LEO, and GEO satellites is set to 100m, 600km, and 35,786km respectively. UEs are distributed around the satellites, UAVs, and eNodeBs in the sub-network.

B. Datasets

As widely used attack datasets in the current cybersecurity field, CICIDS2017 [35], CICIoMT2024 [36], CICIoV2024 [37], and CICIoT2023 [38] collect the attack data for normal network scenarios, healthcare IoT scenarios, automotive scenarios, and IoT scenarios, respectively. However, these datasets only contain single-scenario attack datasets for terrestrial networks, and they lack data related to attacks on UAV networks and satellite networks. As shown in Table III, we compare our dataset (see detailed description in II-D) with the above four datasets, in terms of attacks in SAGINs, including DDoS, DoS, Infiltration, Brute Force, Spoofing, Recon, Web Attack, and Satellite Vulnerability. Our dataset encompasses more diverse attack types and scenarios than existing ones.

C. Baselines

1) Overall Situation Awareness: We compare our method (LLM-SA and NABC fusion), LLM-based methods, and two baseline methods with expert judgment. The two methods are deep autoencoder-deep neural network (AEDNN) based method [28] and network attack behavior classification (NABC) based method [33] respectively use neural networks to classify attacks, and comprehensively consider the occurrence probability and impact of various attacks to give network security situation scores.

2) Attack Correlation Algorithms: The best model and algorithm for computing semantic and feature similarity will be selected based on experimental results. For semantic similarity, we consider BERT [39], Sentence-BERT [40], and SimCSE [41], each of which extends the original BERT architecture to produce context-aware embeddings that effectively capture text similarity. For feature similarity, we evaluate

[3]Llama3: https://llama.meta.com/llama3/

TABLE III
COMPARISONS OF DATASETS

Attacks	DDoS&DoS			Infiltration			Brute Force			Spoofing			Recon			Web Attack	Satellite Vulnerability
Scenarios	Space(S)	Air(A)	Ground(G)	S	A	G	S	A	G	S	A	G	S	A	G	G	S
CICIDS2017	×	×	✓	×	×	✓	×	×	✓	×	×	×	×	×	✓	✓	×
CICIoMT2024	×	×	✓	×	×	×	×	×	×	×	×	✓	×	×	✓	×	×
CICIoV2024	×	×	✓	×	×	×	×	×	×	×	×	✓	×	×	×	×	×
CICIoT2023	×	×	✓	×	×	×	×	×	✓	×	×	✓	×	×	✓	✓	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Cosine Similarity, Jaccard Similarity, and Levenshtein Distance. Cosine Similarity computes the cosine of the angle between vector representations of strings, Jaccard Similarity measures the overlap between sets of characters, and Levenshtein Distance quantifies the minimum edits required to convert one string into another.

3) Dynamic LLM-Based Multi-Agent: We compare our dynamic multi-agent LLM with four baseline models, including Random, MetaGPT [42], LLM Debate [43], and MAGIS [44]. The Random method serves as a performance baseline, randomly selecting answers without any specific strategy or intelligence. It establishes a benchmark for comparing the performance of other methods. MetaGPT employs a multi-agent framework that allows various agents to collaborate on the same task, thereby enhancing the efficiency and accuracy of the model.

LLM Debate improves the overall quality of the program by engaging multiple LLMs in structured debates, allowing them to question each other's responses and iteratively refine the responses through this interactive process. MAGIS employs a scoring agent to generate CVSS v3–based evaluations of information and an assurance agent that verifies and approves these evaluations, requiring revisions until they meet the necessary standards.

D. Metrics

1) Overall Situation Awareness: We calculate the mean squared error (MSE) and root mean squared error (RMSE) in cybersecurity status scores between various methods (including LLM, NABC [33], AEDNN [28], and our method) and expert judgment to reflect the deviation between different methods and expert assessments. MSE and RMSE are standard metrics used to measure the average magnitude of errors between predicted values and observed values. In general, a lower MSE or RMSE indicates that the predictions are closer to the actual observed values, reflecting higher accuracy and precision in the predictions.

2) Threat Information Processing: We calculate the micro-average recall rate as the accuracy for the grouping results, which means dividing the sum of true positives (TP) across all categories by the sum of TP and false negatives (FN) across all categories, reflecting the difference between the grouping results of various methods and the grouping results of experts.

3) Dynamic LLM-Based Multi-Agent: We evaluate the performance of the Dynamic LLM-based multi-agent using two widely-used metrics: relative accuracy and strategy accuracy. Relative accuracy is a comparative metric used to evaluate the precision of various measurement or prediction methods against a known accurate standard. In general, a higher Relative accuracy indicates that the method is closer in precision to the standard, suggesting better reliability and performance. Strategy accuracy is calculated by requiring the model to select one out of four options, where only one option represents the correct standard strategy answer while the other three options are confusing or incorrect strategy answers. It serves as a measure of the correctness of the model's choices.

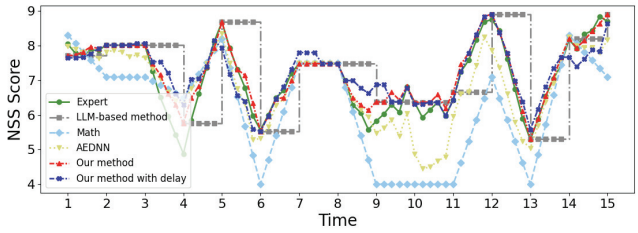


Fig. 4. LLM-based network security situation (NSS) score compared with traditional methods.

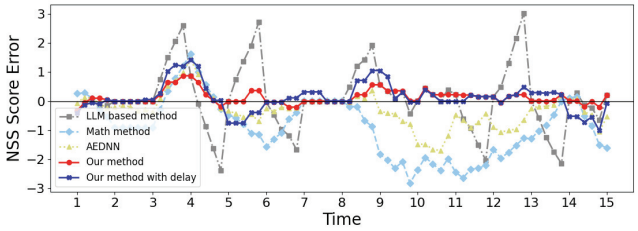


Fig. 5. LLM-based NSS score error compared with traditional methods.

TABLE IV
COMPARISONS OF NSS SCORE

Method	MSE	RMSE
LLM-based method	1.18	1.08
Math	1.72	1.31
AEDNN	0.40	0.63
Our method	**0.07**	**0.27**
Our method with delay	0.24	0.49
Our method with long interval	0.31	0.55

E. Performance

1) Overall Situation Awareness: Figure 4 depicts the temporal evolution of network security situation scores obtained from our proposed method, expert judgment, the AEDNN-based method, the NABC-based evaluation method, and the LLM-based method. The network security situation scores

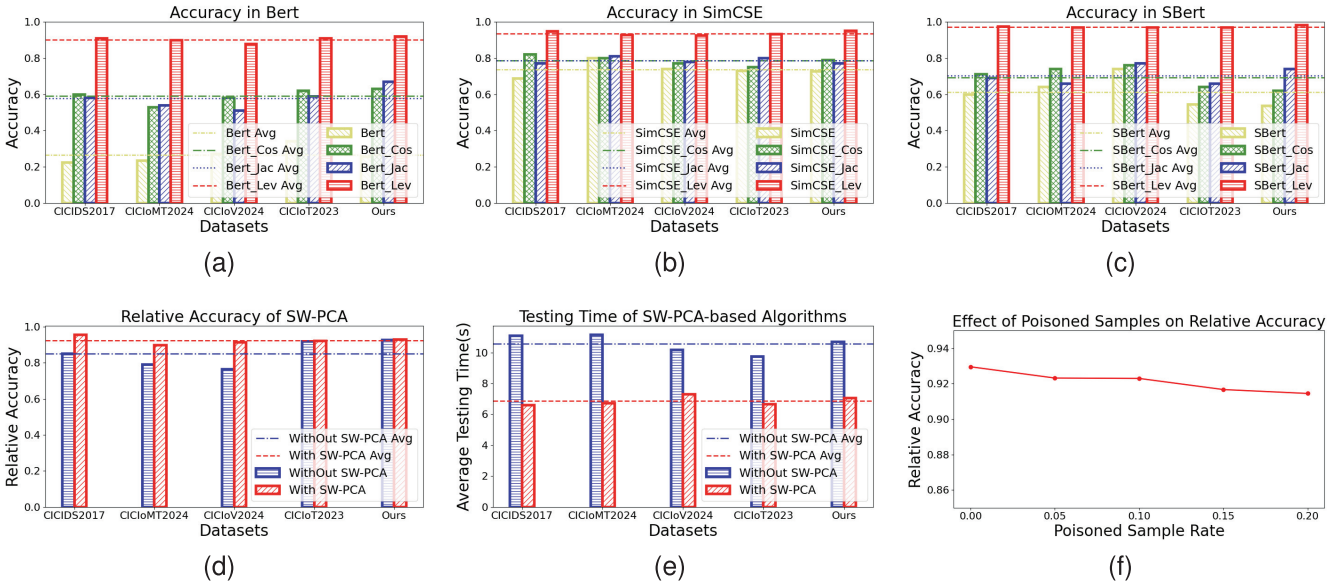


Fig. 6. Performance comparisons on five datasets. (a), (b) and (c) demonstrate the comparisons of the accuracy of the attack correlation algorithms. (d) is the comparison of with and without SW-PCA, (e) indicates the performance among SW-PCA, and (f) shows the robustness of our model with poisoned samples.

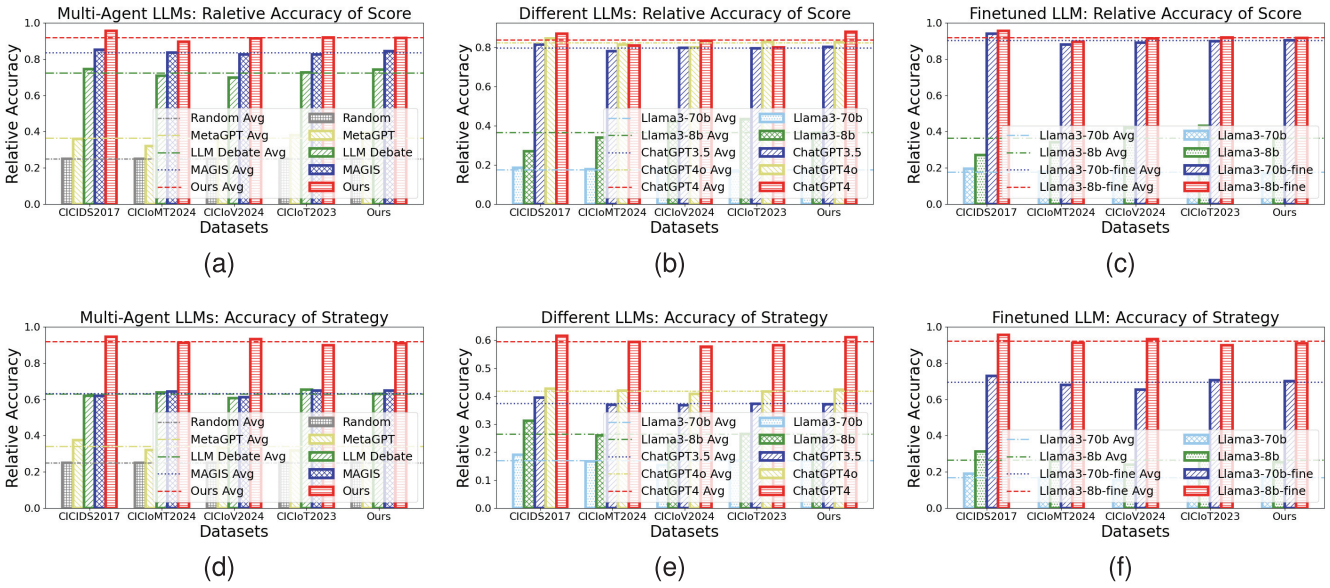


Fig. 7. Performance comparisons on different datasets. (a) and (d) are the comparisons of score and strategy accuracy with different LLM-based multi-agent architectures, (b) and (e) indicate the comparisons among different types of LLMs, (c) and (f) demonstrate the accuracy of the fine-tuned model versus the unfine-tuned model.

are mainly distributed between 4 and 9. Figure 5 illustrates the deviation of these methods' scores from expert judgment scores over time. It can be observed that the results obtained using our proposed method exhibit the closest resemblance to expert judgment, while the results from the other three methods exhibit greater fluctuations and deviations compared to expert judgment. Table IV, Figures 4 and 5 illustrate the real-time environmental scoring curves of our method under a satellite delay of 100 ms. The close alignment between our method's scores and expert scores suggests that the 100 ms delay has minimal impact on the performance of our approach.

Table IV presents the MSE and RMSE of network security scores between multiple methods and the expert judgment model. These error values are significantly lower for our method compared to other baseline methods, indicating that our method provides a more accurate and reliable representation of the cybersecurity situation. Table IV also shows the MSE and RMSE of our model in a resource-limited environment with longer update intervals of LLM-SA.

2) Threat Information Processing: Figures 6 (a), (b), and (c) show the experimental results of our Algorithm 1 in terms of semantic similarity model selection and feature matching algorithm selection, tested with datasets from different scenarios. The experimental results show that we use different semantic similarity models and SBert improves the accuracy of SBert for Bert and SimCSE by 7% and 3% respectively. Compared with other feature-matching algorithms, the accuracy of our feature-matching algorithm is improved by at least 24%.

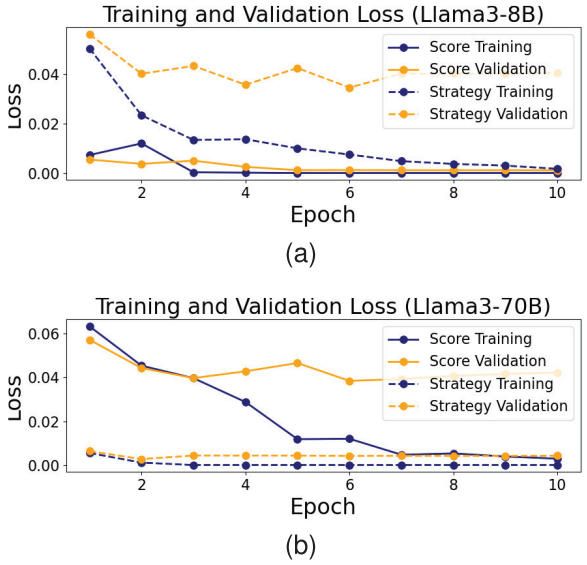


Fig. 8. Training and validation loss over epochs for Llama3-8B (a) and Llama3-70B (b) on scoring and strategy tasks.

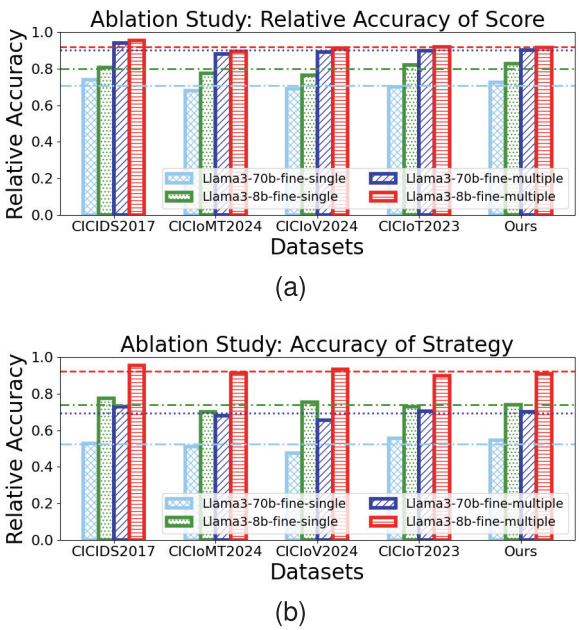


Fig. 9. Ablation study on score relative accuracy (a) and strategy accuracy (b) in a single agent and multiple agents.

In different scenarios, we give the mean line of the algorithm accuracy, and the mean value of our algorithm 1 is at least 27% higher than the other baseline models, which shows that our algorithm has good stability.

We assess the effectiveness of incorporating an SW-PCA extraction algorithm into dynamic LLM-based multi-agent systems by comparing their accuracy and processing times. According to Figures 6 (d) and (e), employing SW-PCA enhances accuracy by 6.5% and reduces processing time by approximately 5 seconds. The SW-PCA method effectively eliminates irrelevant and redundant data while preserving essential information, thereby sharpening the focus and precision of vector database queries by mitigating redundant interference. Consequently, this leads to faster searches and decreased processing times, highlighting the significant benefits of SW-PCA in terms of both accuracy and time efficiency. Figure 6 (f) depicts our model's accuracy across various proportions of poisoned samples, demonstrating minimal loss of accuracy under different intensities of poisoning attacks. This underscores the robustness of our proposed model against such attacks.

3) Dynamic LLM-Based Multi-Agent: To demonstrate the superiority of our dynamic LLM-based multi-agent architecture, we compare its performance against other multi-agent LLM architectures, as shown in Figures 7 (a) and (d). MetaGPT uses a multi-agent framework that enables agents to collaborate on tasks, improving efficiency and accuracy. Meanwhile, LLM Debate raises quality by engaging multiple language models in structured debates, challenging each other's responses, and iteratively refining answers through interaction. MAGIS introduces a scoring agent to generate assessments and an assurance agent that reviews and validates these evaluations, ensuring that only those meeting the required standards are accepted. However, these architectures are tailored to specific tasks. Although cooperative mechanisms work effectively, they may not perform well in the SAGIN scenario, where threat information is complex and heterogeneous. The experimental results show that our architecture achieves high accuracy, surpassing other architectures. This advantage is gained from our dynamic agent generation, which adapts agents based on the prompt context. Additionally, our architecture enhances problem-solving specificity: the Prompt Agent segments information, Specific Advice Agents provide targeted strategies and scores, and the Comprehensive Decision Agent synthesizes the final decision. These two advantages together lead to the achievement of high accuracy.

Figures 7 (b) and (c) demonstrate the impact of varying parameter counts on LLM performance. Our architecture allows for adjustments in LLM types, revealing a positive correlation between increased parameters and accuracy. Notably, ChatGPT-3.5, an online model, exhibits longer response times compared to locally-hosted models. Conversely, models with fewer parameters achieve faster inference times but at the cost of reduced performance. This analysis underscores the trade-off between parameter size and computational intelligence, with intelligence playing a critical role in governing data analysis and tool functions, directly influencing overall accuracy. Furthermore, local deployment of LLMs enhances operational speeds system stability, and security, with Llama3 showing superior performance in our evaluations.

We fine-tune the existing Llama-8b model and the Llama-70b model on the corresponding datasets for rating and strategy tasks. We divide the datasets into training and validation sets with a ratio of 8:2, meaning that 80% of the data is used for training, while the remaining 20% is reserved for validation. We train the models for 10 epochs and monitor their performance by plotting Figures 8 (a) and (b). This process allows us to closely observe any signs of overfitting.

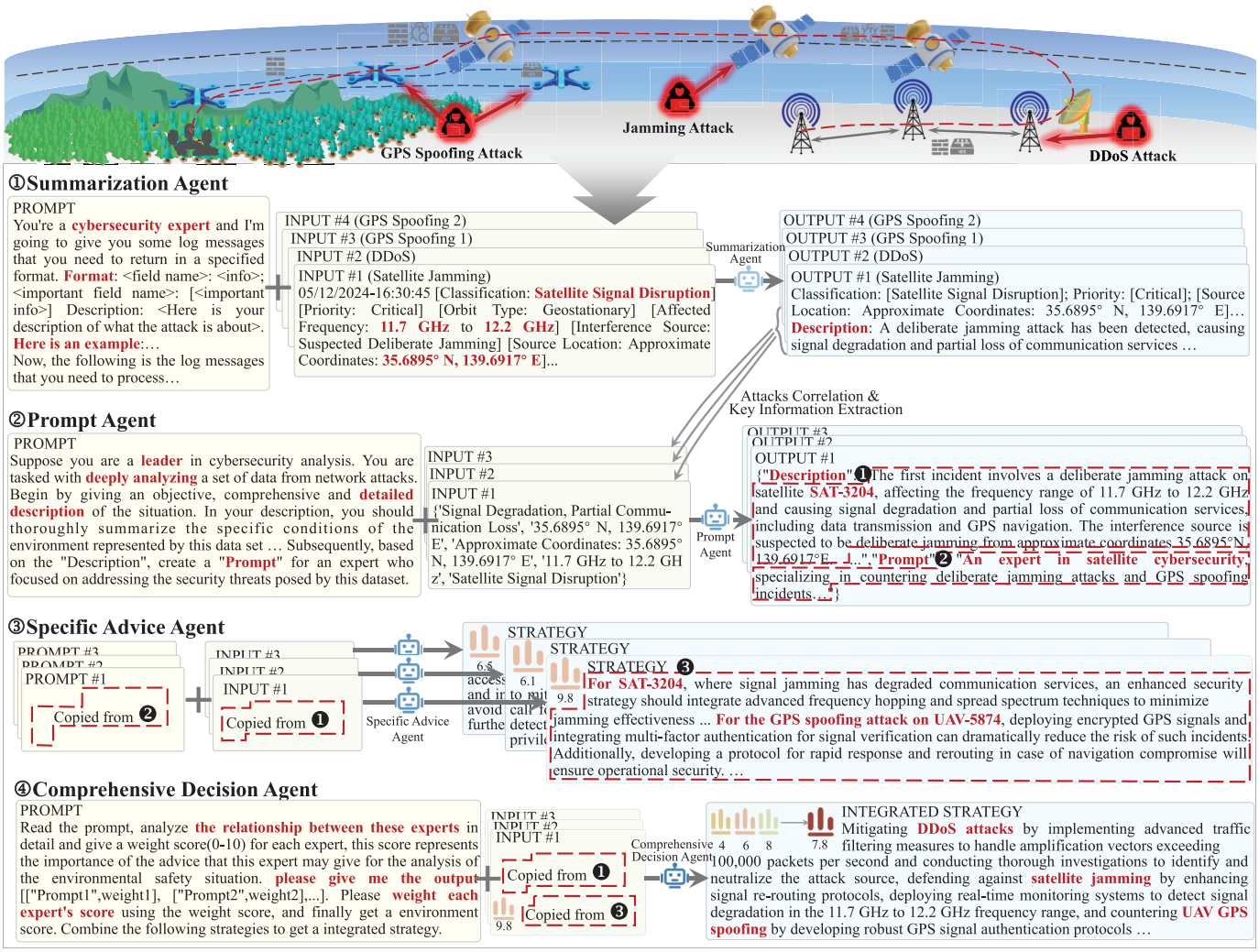


Fig. 10. Case study of LLM-SA. At the top of this figure, an emergency communication scenario is described, which includes three types of attacks: GPS spoofing attack, Jamming attack, and DDoS attack. The workflow at the bottom indicates that our LLM-based method comprises four agents: Information Summarization Agent, Prompt Agent, Specific Advice Agent, and Comprehensive Decision Agent.

TABLE V
COMPARISON OF DIFFERENT LLM PERFORMANCE

Model	Number of Parameters	Score RMSE	Strategy Accuracy	Time cost(s)
ChatGPT-3.5	175B	2.20	0.38	15.31
ChatGPT-4o	-	1.83	0.42	16.54
ChatGPT-4	-	1.49	0.60	21.96
Fine-tuned Llama3	70B	1.11	0.71	91.40
	8B	**0.97**	**0.94**	**7.06**

Figures 7 (c), (f), and Table V show the performance comparison of LLMs before and after fine-tuning. The results show that our fine-tuned Llama-8b model and Llama-70b model achieve improvements of 55% and 74% in relative accuracy, and 66% and 53% in strategy accuracy, respectively. Interestingly, the fine-tuned Llama-8B outperforms Llama-70B in both metrics. Because in scenarios with limited data or lower task complexity, smaller models often exhibit higher parameter efficiency and can outperform larger models. Larger models, despite their strong representational capabilities, are prone to overfitting and require complex optimization strategies. Consequently, smaller models can accelerate training speed and enhance generalization performance. However, excessively small models may lack the necessary capabilities to effectively utilize tools or understand text. Therefore, matching model size to task requirements is crucial for optimal performance. Next, we use the idea of ablation experiments to demonstrate the advancement of the dynamic multi-agent architecture we proposed. We remove the dynamics from the multi-agent architecture, that is, integrate the functions of Prompt Agent and Specific Advice Agent into Comprehensive Decision Agent so that Comprehensive Decision Agent directly receives the output of Algorithm 2 and outputs Integrated Strategy and Total Score. Figures 9 (a) and (b) show that if the agent architecture loses its dynamics, the performance will degrade to varying degrees.

F. Case Study

We consider an emergency communication scenario where UAVs collect information from an area affected by communication interruptions and upload it to the LEO satellite.

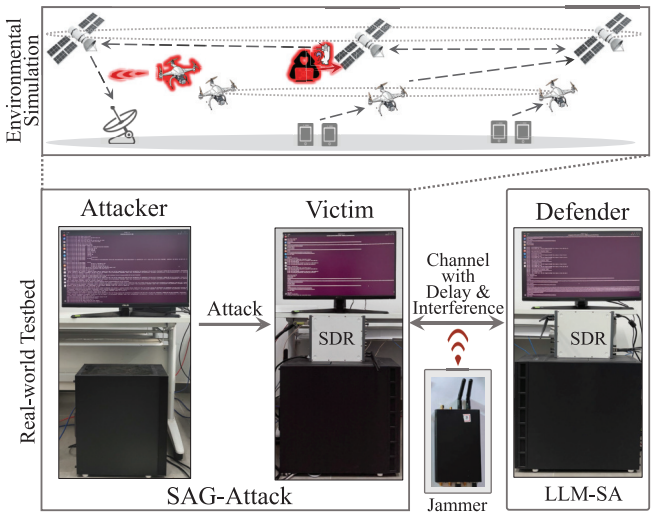


Fig. 11. Illustration of the semi-physical system of SAGIN networks: It consists of a real-world radio system and large-scale simulated non-terrestrial networks. Three servers are an attacker, a victim, and a defender, respectively, where the former two rely on the proposed SAG-Attack platform, and the last one is empowered by LLM-SA. We configure the system with well-known parameters of non-terrestrial networks.

Through inter-satellite and satellite-to-ground communication channels, the information is relayed back to ground stations and subsequently transmitted to ground command centers. Potential cyber threats, such as DDoS attacks, GPS spoofing, and satellite jamming, are identified by security detection tools, which are uploaded to the Summarization Agent.

As shown in Figure 10, the Summarization Agent can understand these unstructured pieces of information (e.g., jamming attacks targeting satellites), synthesize them, and generate key features (e.g., source location) and corresponding descriptions. We perform attack correlation and key information extraction on the four output pieces of information from the Summarization Agent, categorizing two GPS spoofing attacks into one class, ultimately resulting in three distinct lists of key attack information. Using the Prompt Agent, we generate descriptions and prompts for each category of information list. By dynamically specifying the identity of the Specific Advice Agent, we evaluate each category of key information, achieving an adaptive security assessment. Each Specific Advice Agent focuses solely on a specific piece of key information without considering other redundant information and uses its chain-of-thought capability to generate in-depth security strategies. Finally, the Comprehensive Decision Agent considers the number and severity of each type of attack and weighs the environmental scores from each Specific Advice Agent to produce a total score. Additionally, it synthesizes the individual strategies to derive an integrated strategy.

G. Real-World Tests of the Proposed LLM-SA

As shown in Figure 11, we rely on publicly available datasets [45], [46], [47] that include SAGIN network interference and delay to establish an SDR-based Open Air Interface (OAI)[4] wireless communication system for simulating the SAGIN environment. It consists of three servers and two Software-Defined Radios (SDRs). Two servers function as the attacker and the victim, respectively, while the LLM-SA is deployed on a third server running Ubuntu 20.04.4. This server is equipped with 33 GB of RAM, an Intel(R) Core(TM) i9-12900 processor (12th generation), and an NVIDIA GeForce RTX 3060 GPU. The two SDRs facilitate communication between the victim system and the LLM-SA server.

To evaluate the performance of our method in a real-world environment, we implement three types of attacks, including DoS, spoofing, and web attacks on the testbed. We observe that the performance of security assessment and strategies decreases from 91.70% and 90.91% to 89.36% and 90.52% in the real-world tests, representing a reduction of 2.34% and 0.39% compared with the simulation environment, respectively. We calculate the response time from the initiation of each attack to the activation of our security policy. The average response time increases from 7.06s to 8.88s compared with the simulation environment. The experimental results indicate that the model's accuracy experiences a slight decrease, while the response time increases in real-world deployment tests. To further evaluate the response time of our LLM-SA under interference conditions, we introduce a jammer into the physical deployment environment. The experimental results indicate that the model response time increased from 8.88s to 10.14s in the presence of signal interference.

Furthermore, experiments conducted on three Jetson devices demonstrate that the performance of the proposed LLM-SA method in terms of score and strategy generation is 89.81% and 89.52%, respectively, with a loss of less than 2%. This further demonstrates the effectiveness of the proposed LLM-SA system in real-world scenarios and its feasibility for deployment on satellites or UAVs.

VI. Discussions

So far, we have shown the superiority of our proposed LLM-SA on five benchmarks. In this section, we take a further step to highlight some interesting observations, including response time, generalization of agents, and robustness.

A. Efficiency of Our Proposed LLM-SA

Existing LLM methods significantly harden communication systems by making more comprehensive and in-depth security decision-making. However, it is challenging for an LLM to produce a real-time response. To address this issue, we first quantize and fine-tune Llama3-8B, a more efficient LLM compared to ChatGPT-4. While ChatGPT-4 reaches an accuracy rate of 87%, our method based on Llama3-8B achieves 91.7% accuracy. Additionally, Llama3-8B decreases 68% response time of ChatGPT-4. More comparisons are available in Table V. To further minimize response time and enhance real-time capabilities, we combine LLMs with mathematical methods in Algorithm 3.

B. Generalization of Our Proposed LLM-SA

Our LLM-SA meets six criteria as discussed in Section I-C and we conduct experiments to confirm its generalizability.

[4]Open Air Interface: https://openairinterface.org/

Results in Figures 7 (c) and (f) show that the accuracy of our framework is consistently larger than 90% across five distinct datasets, indicating that our method can be well-adapted to various scenarios.

C. Robustness of Our Proposed LLM-SA

To evaluate the robustness of our proposed LLM-SA, we introduced poisoned samples constituting 20% of the training dataset generated by the SAG-Attack simulator. Under such a setting, Figure 6 (f) shows that the assessment accuracy of the proposed LLM-SA decreased by only 1.5%. This confirms the robustness of our LLM-SA.

VII. Conclusion

This paper studies situation awareness of zero-trust SAGIN. We present SAG-Attack and LLM-SA to promote SAGIN's security. Specifically, the proposed SAG-Attack is a simulator that aims to mimic various attacks on communication traffic across satellites, UAVs, and ground networks. Our LLM-SA is a novel situation awareness method that explores the chain-of-thought capabilities of LLMs to conduct security assessments and generate defense strategies. Toward zero-trust SAGIN in practice, the design principles of the proposed SAG-Attack simulator and LLM-SA method are adaptive, learnable, collaborative, pluggable, large-scale, and efficient so that they can be potentially applied to real-world scenarios. We conduct extensive experiments on four public benchmarks to show the effectiveness of the proposed SAG-Attack and LLM-SA. We also build a more comprehensive dataset on the SAG-Attack simulator, and this dataset can serve as a benchmark for zero-trust SAGIN. In the future, we plan to explore the parallelism of LLMs on resource-constrained edge devices.

Appendix I
Details of CVSS

CVSS [29] is comprised of three metric groups: Base, Temporal, and Environmental. We use the Base metric for scoring. The Base metrics represent the intrinsic qualities of a vulnerability that are constant over time and across user environments. This group is further divided into two sub-groups: Exploitability and Impact. Exploitability metrics are as follows.

Attack Vector (AV): Describes how the vulnerability is exploited. Metric Values: Network (N): The vulnerability is exploitable from remote networks. Adjacent (A): Exploitable only within the same shared physical or logical network. Local (L): Exploitable with local access. Physical (P): Physical interaction is required. Numerical Values: Network (N): 0.85, Adjacent (A): 0.62, Local (L): 0.55, Physical (P): 0.2.

Attack Complexity (AC): Describes the conditions beyond the attacker's control that must exist to exploit the vulnerability. Metric Values: Low (L): No special conditions are required. High (H): Special conditions are required. Numerical Values: Low (L): 0.77, High (H): 0.44.

Privileges Required (PR): Describes the level of privileges an attacker must have to exploit the vulnerability. Metric Values: None (N): No privileges required. Low (L): Low-level privileges required. High (H): High-level privileges required. Numerical Values: None (N): 0.85, Low (L): 0.62 (or 0.68 if Scope / Modified Scope is Changed), High (H): 0.27 (or 0.5 if Scope / Modified Scope is Changed).

User Interaction (UI): Whether a separate user must participate in the exploitation. Metric Values: None (N): No user interaction is required. Required (R): User interaction is required. Numerical Values: None (N): 0.85, Required (R): 0.62.

Scope (S): hether the vulnerability affects resources beyond the security scope. Metric Values: Unchanged (U): No impact on other resources. Changed (C): Impacts resources beyond the intended scope.

Impact metrics are as follows.

Confidentiality (C): Impact on confidentiality of the data. Metric Values: None (N): No impact. Low (L): Some data disclosure. High (H): Complete information disclosure. Numerical Values: None (N): 0. Low (L): 0.22. High (H): 0.56.

Integrity (I): Impact on integrity of the data. Metric Values: None (N): No impact. Low (L): Modification of data is possible. High (H): Complete modification of data. Numerical Values: None (N): 0. Low (L): 0.22. High (H): 0.56.

Availability (A):Impact on availability of the system. Metric Values: None (N): No impact. Low (L): Reduced performance. High (H): Complete shutdown of the system. Numerical Values: None (N): 0. Low (L): 0.22. High (H): 0.56.

Appendix II
Details of NABC

The NABC methodology [33] is employed for detecting and analyzing attack behaviors as part of the network security situational awareness process, which unfolds as follows:

Attack Behavior Detection and Analysis: The NABC framework is initially used to identify and scrutinize various attack behaviors within the network. This step is vital for comprehending the nature and patterns of potential threats.

Quantification of Attack Severity: The NABC methodology calculates the error probability matrix and the corrected number of occurrences for each attack behavior to quantify the severity of attacks. These metrics are then integrated with the specific attack severity factor for each behavior, resulting in a quantitative value that signifies the severity of the attacks.

Attack Impact Quantification: The NABC approach assesses the impact of each attack behavior on the confidentiality, integrity, and availability (CIA triad) of the network. Quantifying the degree of impact of each attack behavior facilitates a clearer understanding of the potential consequences of these threats.

Network Security Situation Value Quantitative Calculation: By amalgamating the quantified values of attack severity and attack impact, the NABC methodology derives a comprehensive network security situation value. This value numerically represents the overall security posture of the network.

Network Security Situation Assessment: Finally, the NABC methodology conducts a network security situation

assessment. The evaluation level of the network security situation is determined based on the interval of the network security situation quantification value. This assessment enables the categorization of the network's security posture into predefined levels, supporting informed decision-making and the formulation of appropriate response strategies.

REFERENCES

[1] M. Jia, J. Wu, Q. Guo, and Y. Yang, "Service-oriented SAGIN with pervasive intelligence for resource-constrained users," *IEEE Netw.*, vol. 38, no. 2, pp. 79–86, Mar. 2024.

[2] S. Yao, J. Guan, Y. Wu, K. Xu, and M. Xu, "Toward secure and lightweight access authentication in SAGINs," *IEEE Wireless Commun.*, vol. 27, no. 6, pp. 75–81, Dec. 2020.

[3] H. Dong, C. Hua, L. Liu, W. Xu, and S. Guo, "Optimization-driven DRL-based joint beamformer design for IRS-aided ITSN against smart jamming attacks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 667–682, Jan. 2024.

[4] Y. Sun et al., "RIS-assisted robust hybrid beamforming against simultaneous jamming and eavesdropping attacks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 9212–9231, Nov. 2022.

[5] M. Hajimaghsoodi and R. Jalili, "RAD: A statistical mechanism based on behavioral analysis for DDoS attack countermeasure," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2732–2745, 2022.

[6] L. Crosara, F. Ardizzon, S. Tomasin, and N. Laurenti, "Worst-case spoofing attack and robust countermeasure in satellite navigation systems," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 2039–2050, 2024.

[7] H. Sathaye, M. Strohmeier, V. Lenders, and A. Ranganathan, "An experimental study of GPS spoofing and takeover attacks on UAVs," in *Proc. USENIX Secur. Symp.*, 2022, pp. 3503–3520.

[8] C. Krieger et al., "Integration of scaled real-world testbeds with digital twins for future AI-enabled 6G networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2023, pp. 2037–2042.

[9] X. Chen, W. Feng, N. Ge, and Y. Zhang, "Zero trust architecture for 6G security," *IEEE Netw.*, vol. 38, no. 4, pp. 224–232, Jul. 2024.

[10] P. Dhiman et al., "A review and comparative analysis of relevant approaches of zero trust network model," *Sensors*, vol. 24, no. 4, p. 1328, Feb. 2024.

[11] Y. Liu, Z. Su, H. Peng, Y. Xiang, W. Wang, and R. Li, "Zero trust-based mobile network security architecture," *IEEE Wireless Commun.*, vol. 31, no. 2, pp. 82–88, Apr. 2024.

[12] J. Guo et al., "TFL-DT: A trust evaluation scheme for federated learning in digital twin for mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3548–3560, Nov. 2023.

[13] Y. Ge and Q. Zhu, "GAZETA: GAme-theoretic ZEro-trust authentication for defense against lateral movement in 5G IoT networks," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 540–554, 2024.

[14] W. Yeoh, M. Liu, M. Shore, and F. Jiang, "Zero trust cybersecurity: Critical success factors and a maturity assessment framework," *Comput. Secur.*, vol. 133, Oct. 2023, Art. no. 103412.

[15] B. Yi, Y. P. Cao, and Y. Song, "Network security risk assessment model based on fuzzy theory," *J. Intell. Fuzzy Syst.*, vol. 38, no. 4, pp. 3921–3928, 2020.

[16] J. Wang, M. Neil, and N. Fenton, "A Bayesian network approach for cybersecurity risk assessment implementing and extending the FAIR model," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101659.

[17] N. Yang, R. Gao, Y. Feng, and H. Su, "Event-triggered impulsive control for complex networks under stochastic deception attacks," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1525–1534, 2024.

[18] B. Chen et al., "A security awareness and protection system for 5G smart healthcare based on zero-trust architecture," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10248–10263, Jul. 2021.

[19] Z. Dai et al., "Research on power mobile Internet security situation awareness model based on zero trust," in *Proc. Int. Conf. Artif. Intell. Secur.* Cham, Switzerland: Springer, Jan. 2022, pp. 507–519.

[20] G. F. Riley and T. R. Henderson, *The NS-3 Network Simulator*. Berlin, Germany: Springer, 2010, pp. 15–34.

[21] Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Interactive Services, News Gathering and Other Broadband Satellite Applications; Part 2: DVB-S2 Extensions (DVB-S2X), Standard ETSI EN 302 307-2 V1.3.1, Jul. 2021.

[22] P. Fu, J. Wu, X. Lin, and A. Shen, "ZTEI: Zero-trust and edge intelligence empowered continuous authentication for satellite networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2022, pp. 2376–2381.

[23] I. A. Ridhawi and M. Aloqaily, "Zero-trust UAV-enabled and DT-supported 6G networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 6171–6176.

[24] Z. Guo, J. Cao, X. Wang, Y. Zhang, B. Niu, and H. Li, "UAVA: Unmanned aerial vehicle assisted vehicular authentication scheme in edge computing networks," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 22091–22106, Jun. 2024.

[25] M. Keshavarz, A. Shamsoshoara, F. Afghah, and J. Ashdown, "A real-time framework for trust monitoring in a network of unmanned aerial vehicles," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 677–682.

[26] Z. Fan, Y. Xiao, A. Nayak, and C. Tan, "An improved network security situation assessment approach in software defined networks," *Peer Peer Netw. Appl.*, vol. 12, no. 2, pp. 295–309, Mar. 2019.

[27] C. Yuxin, Y. Xiaochuan, and T. Ren, "A network security situation assessment model based on GSA-SVM," *J. Air Force Eng. Univ.*, vol. 19, no. 5, pp. 78–83, 2018.

[28] H. Yang, R. Zeng, G. Xu, and L. Zhang, "A network security situation assessment method based on adversarial deep learning," *Appl. Soft Comput.*, vol. 102, Apr. 2021, Art. no. 107096.

[29] F. Klement, W. Liu, and S. Katzenbeisser, "Toward securing the 6G transition: A comprehensive empirical method to analyze threats in O-RAN environments," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 2, pp. 420–431, Feb. 2024.

[30] J. H. Seaton, S. Hounsinou, T. Wood, S. Xu, P. N. Brown, and G. Bloom, "Poster: Toward zero-trust path-aware access control," in *Proc. 27th ACM Symp. Access Control Models Technol.*, Jun. 2022, pp. 267–269.

[31] V. Stafford, "Zero trust architecture," *NIST Special Publication*, vol. 800, p. 207, Aug. 2020.

[32] Y. Huang et al., "MVP-tuning: Multi-view knowledge retrieval with prompt tuning for commonsense reasoning," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 13417–13432.

[33] H. Yang, Z. Zhang, L. Xie, and L. Zhang, "Network security situation assessment with network attack behavior classification," *Int. J. Intell. Syst.*, vol. 37, no. 10, pp. 6909–6927, 2022.

[34] J. E. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021.

[35] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, Jan. 2018.

[36] S. Dadkhah, E. C. P. Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi, and A. Ghorbani, "CICIoMT2024: Attack vectors in healthcare devices—A multi-protocol dataset for assessing IoMT device security," Canadian Inst. Cybersecur., Tech. Rep., 2024.

[37] E. C. P. Neto et al., "CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus," *Internet Things*, vol. 26, Jul. 2024, Art. no. 101209.

[38] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023.

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019, pp. 4171–4186.

[40] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[41] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 6894–6910.

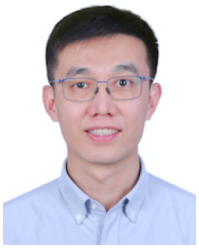
[42] S. Hong et al., "MetaGPT: Meta programming for a multi-agent collaborative framework," in *Proc. 12th Int. Conf. Learn. Represent.*, Jan. 2023.

[43] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," 2023, *arXiv:2305.14325*.

[44] W. Tao, Y. Zhou, W. Zhang, and Y. Cheng, "MAGIS: LLM-based multi-agent framework for GitHub issue resolution," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, Mar. 2024, pp. 51963–51993.

[45] F. Salehi, M. Ozger, and C. Cavdar, "Reliability and delay analysis of 3-Dimensional networks with multi-connectivity: Satellite, HAPs, and cellular communications," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 1, pp. 437–450, Feb. 2024.
[46] J. Zhao and J. Pan, "Low-latency live video streaming over a low-earth-orbit satellite network with DASH," in *Proc. ACM Multimedia Syst. Conf.*, Apr. 2024, pp. 109–120.
[47] N. Mohan et al., "A multifaceted look at starlink performance," in *Proc. ACM Web Conf.*, May 2024, pp. 2723–2734.

Xinye Cao (Graduate Student Member, IEEE) received the B.E. degree from Central China Normal University (CCNU), Wuhan, China, in 2020. She is currently pursuing the Ph.D. degree with the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications (BUPT). Her research interests include large AI model for future wireless communication systems and wireless communications security.



Guoshun Nan (Member, IEEE) is a Professor with Beijing University of Posts and Telecommunications (BUPT). He is a member of the National Engineering Research Center for Mobile Network Technologies. He has published papers in top-tier conferences and journals, including ACL, CVPR, EMNLP, SIGIR, CKIM, SIGCOMM, *IEEE Network*, *Computer Networks*, and *Journal of Network and Computer Applications*. He has broad interests in natural language processing, computer vision, machine learning, and wireless communications, such as information extraction, model robustness, multimodal retrieval, and next generation wireless networks. He served as a reviewer for ACL, EMNLP, AAAI, *Neurocomputing*, and IEEE TRANSACTIONS ON IMAGE PROCESSING.

Hongcan Guo is currently pursuing the bachelor's degree with Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is also a Research Intern with the National Engineering Research Center for Mobile Network Technologies, BUPT. His research interests include post-training of large language models, efficient fine-tuning, reinforcement learning, and mixture of experts models.

Hanqing Mu received the B.E. degree from Beijing University of Technology (BJUT), Beijing, China, in 2023. He is currently pursuing the master's degree with Beijing University of Posts and Telecommunications (BUPT), Beijing. He is with the National Engineering Research Center for Mobile Network Technologies, BUPT. His research interests include AI security, privacy preservation, and their applications in next generation communication systems.

Long Wang received the B.E. degree from the Civil Aviation University of China (CAUC), Tianjin, China, in 2023. He is currently pursuing the master's degree with the National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications (BUPT). His research interests include large AI models, blockchain, and zero-trust security.

Yihan Lin is currently pursuing the bachelor's degree with Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is also a Research Intern with the National Engineering Research Center for Mobile Network Technologies, BUPT. His research interests include wireless communications security and software security.

Qinchuan Zhou is currently pursuing the bachelor's degree with Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is also a Research Intern with the National Engineering Research Center for Mobile Network Technologies, BUPT. His research interests encompass computer vision and large language model-based multi-agent architectures.

Jiayi Li is currently pursuing the bachelor's degree with Beijing University of Posts and Telecommunications (BUPT), Beijing, China. She is also a Research Intern with the National Engineering Research Center for Mobile Network Technologies, BUPT. Her research interests include large language models and deep learning.

Baohua Qin is currently pursuing the bachelor's degree with Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is also a Research Intern with the National Engineering Research Center for Mobile Network Technologies, BUPT. His research interests include deep learning, large language models, and the Internet of Things.

Qimei Cui (Senior Member, IEEE) received the B.E. and M.S. degrees in electronic engineering from Hunan University, Changsha, China, in 2000 and 2003, respectively, and the Ph.D. degree in information and communications engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2006. She has been a Full Professor with the School of Information and Communication Engineering, BUPT, since 2014. She was a Visiting Professor with the Department of Electronic Engineering, University of Notre Dame, IN, USA, in 2016. Her research interests include B5G/6G wireless communications, mobile computing, and the IoT. She won the Best Paper Award at IEEE ISCIT 2012, IEEE WCNC 2014, and WCSP 2019; the Honorable Mention Demo Award at ACM MobiCom 2009; and the Young Scientist Award at URSI GASS 2014. She serves as a Technical Program Chair for APCC 2018, the Track Chair for IEEE/CIC ICCC 2018, and the Workshop Chair for WPMC 2016. She also serves as a Technical Program Committee Member for several international conferences, such as IEEE ICC, IEEE WCNC, IEEE PIMRC, IEEE ICCC, WCSP 2013, and IEEE ISCIT 2012. She serves as an Editor for *Science China Information Science*; and a Guest Editor for *EURASIP Journal on Wireless Communications and Networking*, *International Journal of Distributed Sensor Networks*, and *Journal of Computer Networks and Communications*.

Xiaofeng Tao (Senior Member, IEEE) received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1993, and the M.S. and Ph.D. degrees in telecommunication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1999 and 2002, respectively. He is a Professor with BUPT. He has authored or co-authored over 200 papers and three books in wireless communication areas. He focuses on 5G/B5G research. He is a fellow of the Institution of Engineering and Technology and the Chair of the IEEE ComSoc Beijing Chapter.

He Fang (Member, IEEE) received the Ph.D. degree in applied mathematics from Fujian Normal University, China, in 2018, and the Ph.D. degree in electrical and computer engineering from Western University, Canada, in 2020. She is currently a Full Professor with Fujian Normal University, China. She has over 60 peer-reviewed journals and conference papers. Her research interests include intelligent security provision, trust management, machine learning, distributed optimization, and collaboration techniques. She has received several awards, including the Best Paper Award from IEEE GLOBECOM 2023. She was involved in many IEEE conferences, including IEEE ICC, GLOBECOM, VTC, and ICCC, in different roles such as the Lead Track Chair, the Symposium Chair, the Session Chair, the Workshop Chair, and a TPC Member. She also served as the Vice-Chair for Communication/Broadcasting Chapter, IEEE London Section, Canada, from September 2019 to August 2021. She serves as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and *China Communications*; and a Guest Editor for several journals, including IEEE WIRELESS COMMUNICATIONS and *IEEE Internet of Things Magazine*.

Haitao Du received the Ph.D. degree. He is a Professorate Senior Engineer. He joined China Mobile Research Institute in 2007. He held 29 patents in the area of mobile network security and won seven important science and technology awards. His current research interests include 5G/6G security and security vulnerability mining.

Tony Q. S. Quek (Fellow, IEEE) received the B.E. and M.E. degrees in electrical and electronics engineering from Tokyo Institute of Technology, in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering and computer science from Massachusetts Institute of Technology in 2008. Currently, he is the Cheng Tsang Man Chair Professor with Singapore University of Technology and Design (SUTD) and a ST Engineering Distinguished Professor. He is also the Director of the Future Communications Research and Development Program, the Head of ISTD Pillar, and the AI-on-RAN Working Group Chair in AI-RAN Alliance. His current research topics include wireless communications and networking, network intelligence, non-terrestrial networks, open radio access network, and 6G. He is a fellow of WWRF and the Academy of Engineering Singapore. He was honored with the 2008 Philip Yeo Prize for Outstanding Achievement in Research, the 2012 IEEE William R. Bennett Prize, the 2015 SUTD Outstanding Education Awards-Excellence in Research, the 2016 IEEE Signal Processing Society Young Author Best Paper Award, the 2017 CTTC Early Achievement Award, the 2017 IEEE ComSoc AP Outstanding Paper Award, the 2020 IEEE Communications Society Young Author Best Paper Award, the 2020 IEEE Stephen O. Rice Prize, the 2020 Nokia Visiting Professor, the 2022 IEEE Signal Processing Society Best Paper Award, and the 2024 IIT Bombay International Award For Excellence in Research in Engineering and Technology.