Intelligent Orchestration Method for Cybersecurity Services in Satellite-Terrestrial Integrated Network[*]

Wangjian Zhou, Xiaojing Fan, and Huachun Zhou[†]

Beijing Jiaotong University, Beijing, China
{23125093, 23111043, hchzhou}@bjtu.edu.cn

Abstract

The current mobile communication network still has obvious deficiencies in global land coverage, and traditional terrestrial communications have difficulty in balancing coverage breadth and transmission capacity. Satellite-Terrestrial Integrated Network(STIN) realizes wide-area coverage by integrating satellite and terrestrial communication resources, which has become an important direction for the construction of a new type of network system. However, with the continuous expansion of the network scale, the network faces more complex security threats, and it has become a key challenge to improve the utilization of node resources and reduce the traffic transmission delay. This paper focuses on the intelligent scheduling of security service paths in the STIN, based on a three-layer STIN architecture built upon Software-Defined Networking (SDN), and clarifies the functional division of labor between the ground and multi-layer satellite network nodes. On the basis of the STIN architecture,a knowledge-sharing mechanism between the control and forwarding layers is constructed to realize the sharing of information such as traffic characteristics and the detection capability of security function nodes. Further, this paper proposes a dynamic security service path selection algorithm based on double deep Q-network (DDQN) reinforcement learning. Experimental results show that the method effectively reduces the load of satellite nodes under multiple types of Distributed Denial of Service(DDoS) attacks and normal traffic scenarios. Compared with the baseline path, it reduces the path delay by 49.98%, which significantly improves the security and service efficiency of the STIN.

Keywords: Satellite-Terrestrial Integrated Network; Software-Defined Network; Security Service Function; Path Orchestration

1 Introduction

The Satellite–Terrestrial Integrated Network (STIN), by integrating satellite and terrestrial communication resources, achieves wide-area coverage and efficient services based on a unified protocol stack and intelligent control. It has become a key solution for building a comprehensive, reliable, and secure communication system. At present, global terrestrial mobile communication networks still suffer from significant coverage limitations. As of July 2023, only about 20% of the land area is covered, leaving more than 30% of the global population without reliable Internet access. This highlights the imbalance between coverage and transmission capacity in traditional communication methods [1], and urgently calls for breakthroughs enabled by the new architecture of the satellite–terrestrial integrated network.

For STIN, people not only have to face diverse emerging network services but also stricter security requirements. Since Release 15, the 3rd Generation Partnership Project (3GPP) has started to study and formulate relevant standards and technical frameworks for Non-Terrestrial

[*] Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 60, December 16-18, 2025, Sapporo , Japan.

[†] Corresponding author

Networks (NTN). After Release 17 formally incorporated NTN into the 5G architecture, the need to ensure consistent security with terrestrial networks was proposed for NTN scenarios [2].

However, with the continuous expansion of network scale and the diversification of services, STIN still suffers from limited cross-layer coordination, static allocation of security functions, and insufficient intelligence in orchestration and management [3]. In particular, existing STIN security frameworks often rely on hierarchical yet loosely coupled architectures, where coordination among satellite layers as well as between satellite and terrestrial segments remains inefficient. Furthermore, the static and preconfigured deployment of security functions hinders dynamic adaptation under varying traffic conditions and heterogeneous service demands. These structural limitations, coupled with increasing threats such as Distributed Denial of Service (DDoS) attacks [4], highlight the urgent need for dynamic and adaptive security mechanisms capable of efficient resource utilization and real-time response.

With the development of new network technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV), network control and function deployment are being gradually decoupled from hardware and implemented in a flexible, software-driven manner. NFV enables the virtualization of traditional security functions (e.g., intrusion detection systems and firewalls) into Virtual Network Functions (VNFs) that can be automatically deployed and centrally managed to meet automation and flexibility demands of network services. Based on this, Service Function Chaining (SFC) [5] can orchestrate multiple VNFs according to diverse security requirements and generate customized service paths for specific scenarios. Accordingly, this paper proposes an intelligent orchestration method of security service functions for STIN scenarios. The method builds a security service–oriented architecture based on SDN/NFV technologies, deploys VNFs to offer diversified security services, and introduces deep reinforcement learning to perform intelligent scheduling. Ultimately, scheduling paths are enforced via the SFC mechanism to enable flexible composition and efficient coordination of security functions. This work aims to provide a reference architecture for service-oriented security deployment in STIN.

2 Related Work

In light of the growing prevalence of cyber-attacks, the STIN, recognized as a foundational architecture for next-generation communication systems, has become a focal point of cybersecurity research. A substantial body of work has emerged to address threat detection and mitigation within these environments. Jiang et al. [6] proposed a routing protocol that integrates intrusion detection mechanisms to identify malicious behaviors at network nodes; once detected, traffic is collaboratively rerouted via multiple satellite nodes to bypass compromised paths. Guo et al. [7] introduced a distributed collaborative defense framework based on blockchain, employing service function chains (SFCs) and the MapReduce algorithm together with blockchain technology to record ingress traffic features and aggregate attack signatures, thereby enabling decentralized detection of malicious patterns. To enhance adaptability in dynamic threat scenarios, Deng et al. [8] presented a deep reinforcement learning based solution that flexibly composes security service functions by intelligently combining various detection modules, thus strengthening the adaptability of SFCs against evolving threats. From a network architecture perspective, Li et al. [9] proposed a horizontal multi-domain SFC orchestration framework for SDN-enabled satellite networks, utilizing a heuristic mapping algorithm to coordinate both inter-domain and intra-domain path computation and to effectively deploy service functions across multiple domains.

While these approaches represent significant advancements in areas such as intrusion-aware routing, blockchain-enhanced defense, learning-driven function composition, and multi-domain orchestration, they primarily focus on optimizing individual security mechanisms in isolation. Most lack comprehensive coordination across terrestrial and satellite infrastructures. Furthermore, the complexity and variability of attack vectors, combined with the resource constraints inherent in satellite environments, mean that existing solutions often fall short in terms of security policy flexibility, adaptive response, and efficient cross-domain scheduling. To address these limitations, this paper proposes an intelligent orchestration method for security services in STIN, leveraging SDN and SFC technologies. By introducing centralized SDN control and dynamic SFC-based path orchestration, the proposed approach enables holistic resource integration, real-time service awareness, and efficient responses to DDoS and other cyber-attacks, thereby enhancing the security resilience of the entire network architecture.

3 Research Design and Implementation

In this section, the overall framework architecture for service-oriented intelligent orchestration of security functions is first described based on the SDN three-layer network architecture. Then the process of dynamic sensing of management-level knowledge is explained in detail. Finally, the intelligent decision-making method based on double deep Q-network (DDQN) deep reinforcement learning is presented.

3.1 Overall Scheme

As illustrated in Figure 1, this paper further enhances the service-oriented orchestration capabilities for security functions based on the SDN-enabled three-layer architecture proposed by Li et al. [10]. Specifically, a space traffic generation module is deployed to enable seamless interoperability between the TCP/IP protocol stack and the Delay/Disruption Tolerant Networking (DTN) stack.

In the Medium Earth Orbit (MEO) satellite layer, security function modules are deployed to perform real-time traffic analysis and threat mitigation. A knowledge dynamics awareness mechanism is introduced between the MEO and Geostationary Earth Orbit (GEO) satellite layers. This mechanism facilitates the real time upload of information, including traffic characteristics, attack detection capabilities, and resource utilization metrics, from security function nodes in the MEO layer to a centralized knowledge base maintained on GEO satellites. This enables a holistic understanding of the network status across different orbital layers.

Building upon this, the multidimensional information stored in the GEO-based knowledge base is analyzed using a DDQN reinforcement learning model. The DDQN is responsible for intelligently scheduling security function nodes in the MEO layer and generating optimized service function paths. These paths are then dynamically deployed and executed through a combination of security service controllers and SFC controllers in the SDN control layer, enabling flexible, on-demand security service delivery.

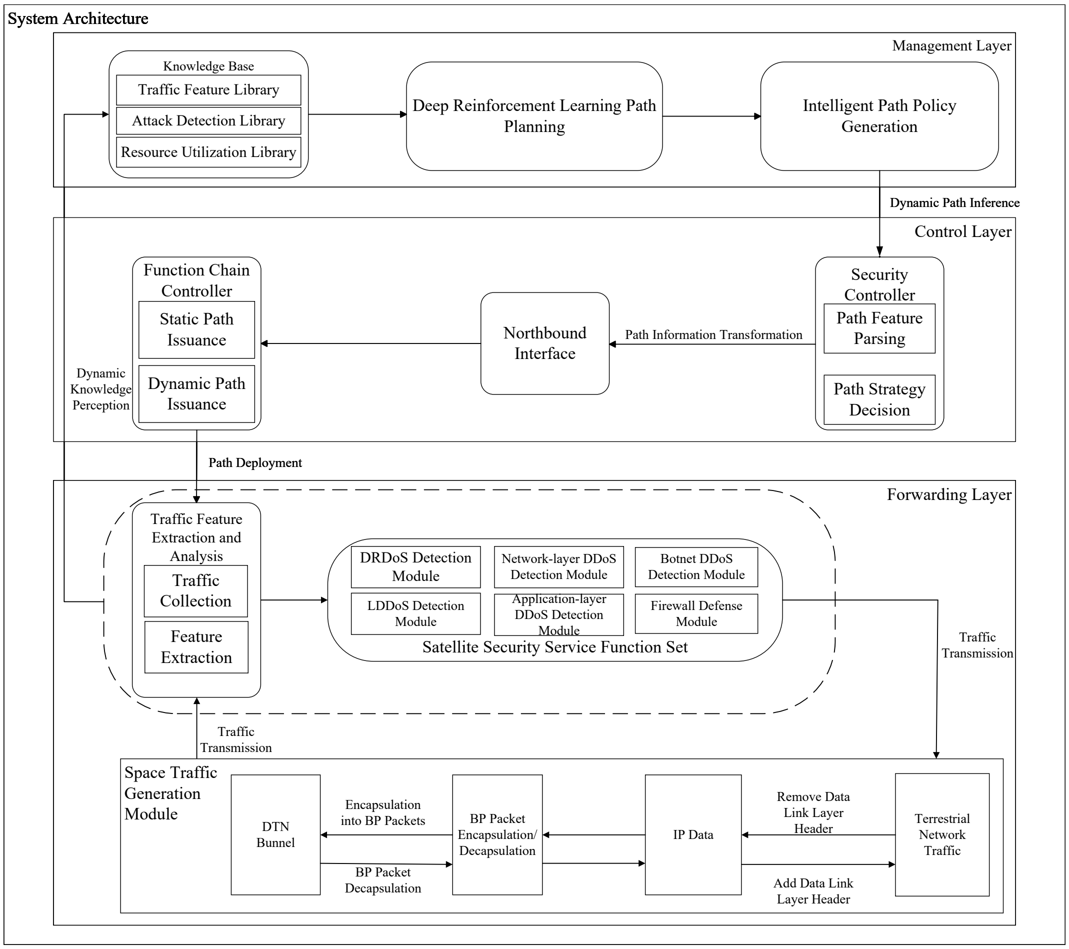

System Architecture
Management Layer
Knowledge Base
Traffic Feature Library
Attack Detection Library
Resource Utilization Library
Deep Reinforcement Learning Path Planning
Intelligent Path Policy Generation
Dynamic Path Inference
Control Layer
Function Chain Controller
Static Path Issuance
Dynamic Path Issuance
Northbound Interface
Path Information Transformation
Security Controller
Path Feature Parsing
Path Strategy Decision
Dynamic Knowledge Perception
Path Deployment
Forwarding Layer
Traffic Feature Extraction and Analysis
Traffic Collection
Feature Extraction
DRDoS Detection Module
Network-layer DDoS Detection Module
Botnet DDoS Detection Module
LDDoS Detection Module
Application-layer DDoS Detection Module
Firewall Defense Module
Satellite Security Service Function Set
Traffic Transmission
Traffic Transmission
Space Traffic Generation Module
DTN Bunnel
Encapsulation into BP Packets
BP Packet Decapsulation
BP Packet Encapsulation/ Decapsulation
IP Data
Remove Data Link Layer Header
Add Data Link Layer Header
Terrestrial Network Traffic

Figure 1: Framework Architecture for Cybersecurity Orchestration in STIN

3.2 Dynamic Knowledge Perception

To provide the management layer with the capability to perceive the traffic entering the security function modules of MEO satellites and to support subsequent adjustments of intelligent orchestration strategies, this paper deploys a traffic feature feedback submodule and a traffic detection capability submodule within the forwarding layer. These components interact with the management layer of GEO satellites to enable dynamic knowledge perception at the GEO management layer.

The Traffic Characterization Feedback submodule is deployed at the ingress point of the SFC and is responsible for collecting traffic data from the MEO satellite within a fixed time window. This process enables the extraction of key features from real-time traffic streams. The submodule captures incoming traffic by monitoring the network interface card (NIC) using the `Tcpdump` tool. The captured traffic is stored in a `.pcap` file within each predefined time window for subsequent feature extraction.

The module traverses all packets within each `.pcap` file to extract a total of 17 traffic features, including metrics such as source IP entropy, destination IP entropy, and other statistical descriptors. These extracted features are structured and stored in JSON format, and subsequently appended to the SFC ingress node for later analysis and processing, as detailed in Table 1.

To support data exchange with the management layer, a NETCONF-based control communication mechanism is implemented over the DTN architecture. This enables the construction of a stable control channel between the MEO and GEO satellites. Through this channel, the extracted traffic features from the space segment are continuously and reliably transmitted to the management-layer knowledge base, thereby providing critical data support for traffic sensing, security monitoring, and policy optimization tasks.

Table 1: Traffic Features

Symbol	**Description**
R_{pkt}	Packet rate (packets per second)
R_{byte}	Byte rate (bytes per second)
$\overline{L}_{\text{pkt}}$	Average packet length (bytes)
H_{sip}	Entropy of source IP addresses
ΔH_{sip}	Change in entropy of source IP addresses
H_{dip}	Entropy of destination IP addresses
ΔH_{dip}	Change in entropy of destination IP addresses
H_{ttl}	Entropy of TTL (Time To Live) values
H_{tcp_sport}	Entropy of TCP source ports
H_{tcp_dport}	Entropy of TCP destination ports
H_{udp_sport}	Entropy of UDP source ports
H_{udp_dport}	Entropy of UDP destination ports
H_{plen}	Entropy of packet lengths
$H(sip\|dip)$	Conditional entropy between source and destination IPs
$H(sip\|dport)$	Conditional entropy between source IP and destination port
$H(dport\|dip)$	Conditional entropy between destination port and destination IP
σ^2_{pkt}	Variance of the number of packets

The detection result feedback submodule is implemented across all security function modules deployed on MEO satellites. These modules are responsible for integrating various DDoS detection mechanisms, including models for Distributed Reflection Denial of Service (DRDoS), Low rate DDoS (LDDoS), application layer DDoS, network layer DDoS, and traditional firewall policies.

When traffic traverses the security function nodes in the space network, each detection model processes the corresponding flow and generates detection results. These results are locally stored and concurrently transmitted to the management-layer knowledge base via the NETCONF protocol. This facilitates the real-time upload and sharing of flow-level detection outcomes, supporting comprehensive situational awareness and adaptive decision-making in network security management.

3.3 Intelligent Path Generation

In GEO satellites, a DDQN-based deep reinforcement learning algorithm is employed to intelligently orchestrate security service functions within the STIN. The orchestration process is modeled as a Markov Decision Process (MDP). By constructing both an evaluation network and a target network, the system dynamically schedules paths for SFCs deployed across the satellite network.

The proposed DDQN-based path orchestration framework is illustrated in Figure 2. As detailed in Section 3.2, upon the completion of knowledge interaction, the knowledge base located on the GEO satellite receives traffic feature information and detection capabilities from the MEO security submodules. This data is stored in both the *Traffic Detection Database* and the *Resource Usage Database*.

Simultaneously, the MEO satellites analyze real time network traffic using built in security monitoring mechanisms, extracting key traffic characteristics and transmitting them back to the GEO satellite as part of the system's environmental state.Upon receiving this status feedback, the GEO satellite integrates the data with its existing knowledge base, traffic feature database, and resource usage database to construct a comprehensive representation of the current environment. This representation serves as the input to the DDQN-based decision-making process for intelligent service function orchestration.

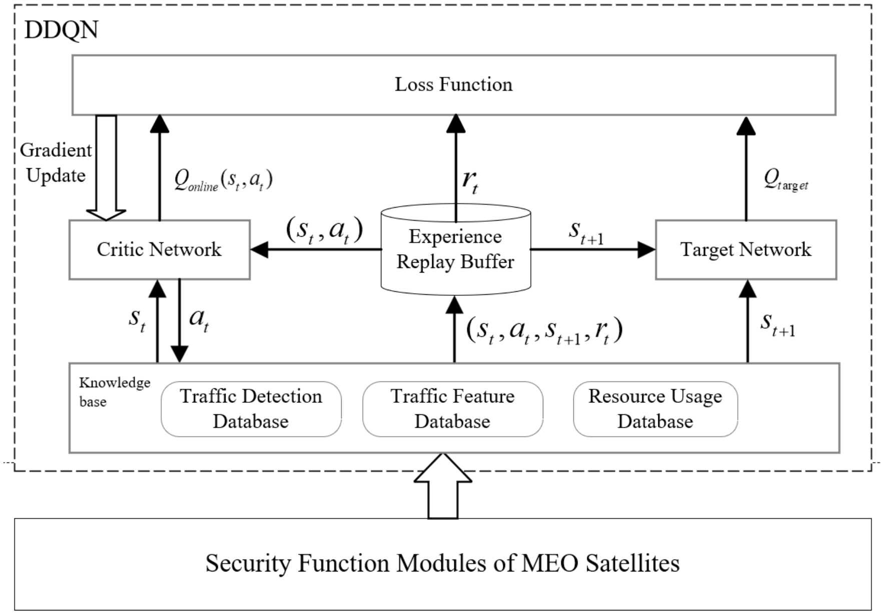


Figure 2: Intelligent Path Generation

In the Intelligent Path Generation, the state space S is:

$$S = \{s_t \mid s_t \in [f_1^{(i)}, f_2^{(i)}, \ldots, f_{35}^{(i)}]\} \tag{1}$$

The state space is composed of two parts: the flow characteristics and the function module flag bits, which contains a total of 35 dimensions. The first 17 dimensions are used to represent the flow characteristics of the current flow, and the last 18 dimensions are designed as the function flag bits, with each function node corresponding to 3 flag dimensions. Based on this, the evaluation network receives the environmental state S_t from the state space S, and estimates the Q-values of all candidate actions.

Action space for:

$$A = \{a_t \mid a_t \in PathSet\} \tag{2}$$

Where Pathset is an array of 0-5 representing six discrete actions corresponding to five MEO satellite safety function modules as well as one firewall module. Based on this, the evaluation network receives the environmental state S_t from the state space S, and estimates the Q-values of all candidate actions.

The environmental feedback bonus value is defined as the ratio between the detection module corresponding to the selected action and the overall path length. For abnormal situations that occur within the service path, since they cannot further enhance the overall detection capability and instead lead to additional resource consumption, negative feedback is applied to suppress such behaviors. After multiple rounds of experimental validation, the parameter settings are determined as shown in the following equation:

$$r_t = \begin{cases} -150, & \textit{Repeated Action} \\ -100, & \textit{Reasoning path length greater than 6} \\ \frac{200}{\mathrm{step}} + \varepsilon, & \textit{Select Firewall} \\ -500 + \varepsilon, & \textit{feature repetition} \\ \frac{\mu - 100}{\mathrm{step}} + \varepsilon, & \textit{else} \end{cases} \tag{3}$$

Here, *step* represents the length of the inferred path, and μ denotes the resource utilization rate in the knowledge base. ε is a random value within the range of $[-20, 20]$, used to introduce stochastic disturbance and enhance exploration randomness during policy search. This randomness enables exploratory samples to be added to the experience replay buffer.

Meanwhile, the DDQN algorithm employs a combination of an evaluation network and a target network. The target network is responsible for computing the target Q-values, while the evaluation network estimates the Q-values corresponding to the current policy. By minimizing the difference between the target Q-values and the estimated Q-values, the system continuously optimizes the loss function and updates the weights of the evaluation network. The target network, on the other hand, periodically copies the parameters from the evaluation network to mitigate the problem of overestimation during Q-value updates.

Algorithm 1 demonstrates the offline training process based on DDQN. During training, the intelligent agent interacts with the state and action spaces in a customized reinforcement learning environment. The action space consists of six types of discrete network security functions (e.g., DRDoS, network layer DDoS, Botnet, etc.), where each action is mapped to a corresponding functional node. The state is represented by a 35-dimensional feature vector that includes both traffic characteristics and historical response information.

The environment dynamically updates the state based on the outcome of action execution and applies reward and penalty mechanisms based on system resource consumption and path redundancy, thereby enabling dynamic decision feedback. An experience replay buffer is constructed using historical feature data as training input. Each experience sample consists of a tuple$(\mathbf{s}, a, r, \mathbf{s}', \mathrm{done})$, where $\mathbf{s}$ is the current state, a is the action, r is the reward, $\mathbf{s}'$ is the next state, and done is the episode termination flag.

At each iteration, the agent selects an action using an ε-greedy strategy, computes the Q-value of the current state using an evaluation network, and estimates the target Q-value of the next state using a target network. Unlike the standard DQN, DDQN mitigates the overestimation of Q-values by decoupling action selection from value evaluation. The target network parameters are periodically synchronized with those of the evaluation network to maintain training stability.

This training process is continuously sampled and updated until a neural network model is learned that can dynamically select the optimal secure path. The final trained model is deployed on a GEO satellite to perform path inference on real-time network traffic characterization data.

Algorithm 1 Offline Training Process Based on DDQN

Input: Traffic feature set $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$, initialized evaluation network $Q_{\text{online}}(\theta)$, target network $Q_{\text{target}}(\theta')$, replay memory with capacity M

Output: Trained neural network model for dynamic path selection

```
Initialize DDQN model, replay memory $M$, $Q_\theta$, and $Q_{\theta'}$
Initialize action space $A = \{0, 1, 2, 3, 4, 5\}$ and action-to-node mapping
for episode = 1 to $N$ do
    Reset environment:
        Load latest traffic feature file
        Extract latest row to form feature vector $\mathbf{f} \in R^{17}$
        Initialize state $\mathbf{s} = [f, 0, \ldots, 0] \in R^{35}$
        Reset step counter $\leftarrow 0$, path choice $\leftarrow$
        $r_{\text{episode}} \leftarrow 0$, done $\leftarrow$ False
    while not done do
        Choose action $a$ using $\varepsilon$-greedy strategy:
            With probability $\varepsilon$: $a \leftarrow \text{random}(A)$
            Otherwise: $a \leftarrow \arg\max_a Q_\theta(\mathbf{s}, a)$
        Execute action $a$ in environment:
            Append node to path if not already present
            Update state $\mathbf{s}'$ based on logic
            Compute reward $r$ using Eq. (2)
        if step counter $> 7$ or $a = \mathit{firewall}$ then
            done $\leftarrow$ True
        end if
        Store transition $(\mathbf{s}, a, r, \mathbf{s}')$ into memory $M$
        $\mathbf{s} \leftarrow \mathbf{s}'$, $r_{\text{episode}} \leftarrow r_{\text{episode}} + r$
    end while
    if memory $M$ is full then
        Sample minibatch of transitions from $M$
        for each transition $(\mathbf{s}, a, r, \mathbf{s}')$ do
            $a' \leftarrow \arg\max_a Q_\theta(\mathbf{s}', a)$                    ▷ Action selection
            $Q_{\text{target}} \leftarrow r + \gamma \cdot Q_{\theta'}(\mathbf{s}', a')$                ▷ Target Q-value
            Update $Q_\theta$ by minimizing loss between predicted $Q_\theta(\mathbf{s}, a)$ and $Q_{\text{target}}$
        end for
    end if
    if episode % $K = 0$ then
        Update $Q_{\theta'} \leftarrow Q_\theta$
    end if
    Record $r_{\text{episode}}$ and training loss
end for
```

4 Experiments

This experiment simulates legitimate traffic and multiple types of DDoS attacks in 5G multi-scenarios in a STIN environment to evaluate the performance of the proposed deep reinforcement learning model and to validate the path inference function and its scheduling effect.

4.1 Experimental Environment

This paper develops a multi-layer satellite network (MLSN) simulation environment based on the VMware *vSphere* virtualization platform. The network consists of 3 GEO, 10 MEO, and 66 Low Earth Orbit (LEO) satellites, where all satellite nodes are configured to support the DTN protocol stack.

In this architecture, the knowledge base and the DDQN deep reinforcement learning model are deployed on the GEO satellites. The MEO satellites serve as security function nodes responsible for traffic forwarding, threat identification, and malicious traffic interception. The LEO satellites primarily support traffic forwarding tasks.

As illustrated in Figure 3, the experimental network topology is designed accordingly. The IP addresses of the attacking hosts are:

- 23.1.0.1, 23.1.0.7, 23.1.0.4, 10.1.0.1, 23.1.0.6

The IP addresses of the attack target nodes are:

- 23.1.1.6 and 10.1.1.1

In the figure, the red curves denote the original service paths predetermined by the system, which traverse all MEO satellite-based security function nodes. In contrast, the blue path indicates the optimal service route dynamically selected by the system based on intelligent decision-making when subjected to LDDoS attacks, application-layer DDoS attacks, and Botnet hybrid attacks.

The legitimate hosts that generate normal traffic use virtual IP addresses in the range 23.1.1.6 to 23.1.1.13. Both benign and malicious traffic originates from the ground. When traversing the terrestrial–satellite gateway, a protocol stack conversion between TCP/IP and DTN is executed to ensure effective transmission into the satellite network.

Simultaneously, real-time traffic feature collection, resource utilization monitoring of the MEO-based security function nodes, and their attack detection capabilities are transmitted to the knowledge base on the GEO satellite. The DDQN model deployed on the GEO satellite then dynamically performs intelligent orchestration of security service functions in response to real-time traffic conditions.

4.2 Evaluation Metrics

In this paper, we employ four evaluation metrics: accuracy, precision, recall, and F1-score, to comprehensively evaluate the performance of intelligent security SFCs in detecting malicious traffic in a mixed-traffic environment.

For the evaluation of path performance in terms of delay and load, the following calculation method is adopted:

The path delay is obtained by summing the detection processing time of each security function node along the path and the forwarding delay between nodes. The average delay over all sampled traffic is then computed using the arithmetic mean:

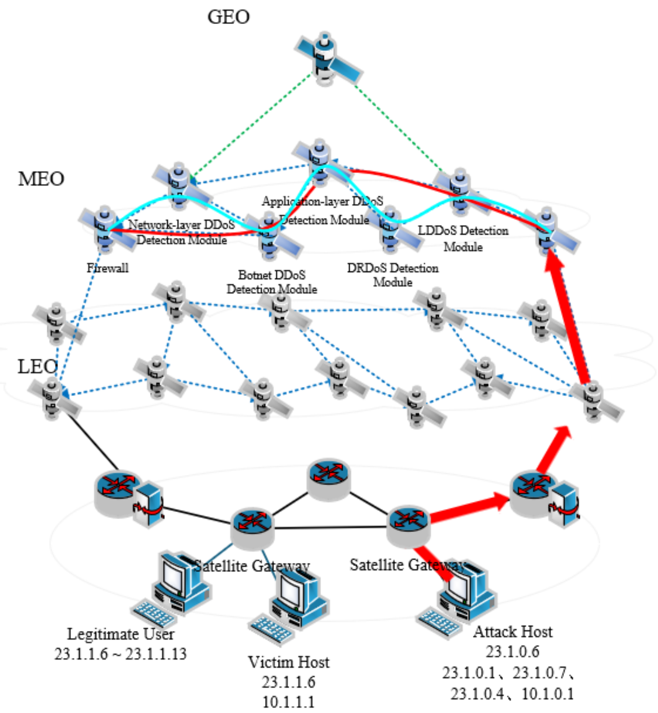


Figure 3: Experimental Environment

$$AvgDelay = \frac{1}{N}\sum_{i=1}^{N}\left(D_i^{fwd} + D_i^{det}\right) \tag{4}$$

where D_i^{fwd} denotes the transmission delay, D_i^{det} represents the detection delay at the i-th security function node, and N denotes the total number of function nodes along the path.

Path load, on the other hand, is calculated by measuring the CPU utilization of each security function node in a path during a fixed time window, and then weighting, summing, and averaging these values as a measure of the path's resource consumption in handling a specific traffic flow:

$$AvgLoad = \frac{1}{T \cdot N}\sum_{i=1}^{N}\sum_{t=1}^{T} Load_{i,t} \tag{5}$$

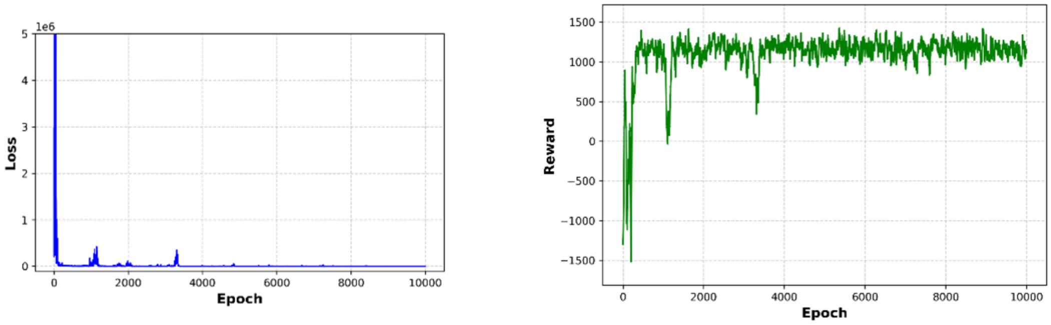
where T denotes the length of the time window, N denotes the number of functional nodes in the path, and $Load_{i,t}$ represents the CPU occupancy of the i-th functional node at the upper edge of time t.

4.3 DDQN-Based Security Function Orchestration

4.3.1 offline training

In the offline training phase, the model is implemented using Python 3.6 and PyTorch 1.6 to construct the deep neural network. During the experiments, traffic generated by DRDoS attacks, network-layer DDoS attacks, LDDOS attacks, application-layer DDoS attacks, botnet attacks, and mixed DDoS attacks is sent to the MEO satellite security function modules. Simultaneously, the traffic characteristics and attack detection results of each MEO satellite security function module are stored in the GEO satellite knowledge base via inter-satellite knowledge exchange. The knowledge stored in the knowledge base is subsequently employed to train the DDQN model.

The reward and loss values are further used to evaluate the training effectiveness of the proposed model, as illustrated in Figure 4. As shown in Figure 4(a), the loss value exhibits a gradual decrease with an increasing number of training iterations, indicating effective model convergence. Figure 4(b) demonstrates that the path reward value generally increases as training progresses. However, between 3000 and 4000 training steps, noticeable fluctuations in reward values are observed despite the overall upward trend. After approximately 8000 training iterations, the reward value stabilizes, suggesting that the agent has converged to a relatively optimal decision-making policy.



(a) Loss

(b) Reward

Figure 4: Offline Training Metrics

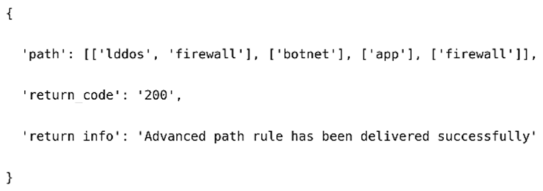
4.3.2 Function Verification

In this paper, the initial network path is preconfigured to traverse all security function nodes deployed on the MEO satellites. This default path ensures that all incoming traffic initially passes through each security function node sequentially. During the experiment, the terrestrial network generates and transmits various types of data to the satellite network, including five categories of typical DDoS attack traffic and legitimate traffic.

Upon arrival at the MEO satellites, the system interacts with the GEO satellites, which host the knowledge base and the DDQN model, to analyze the traffic and intelligently schedule security service functions. Customized service paths are then dynamically generated according to the nature of the traffic and the available system resources.

As illustrated in Figure 5, when the incoming traffic comprises a mixture of LDDoS attacks, a small number of Botnet attacks, application layer DDoS attacks, and normal traffic, the orchestrated paths selectively include only the security function nodes and firewall components capable of detecting and handling these specific types of attacks.

Notably, for traffic identified as LDDoS by the corresponding detection module, the optimized path consisting of the *lddos-firewall* chain enables rapid forwarding to the firewall node for immediate interception, thereby enhancing processing efficiency and minimizing resource overhead.



```
{
  'path': [['lddos', 'firewall'], ['botnet'], ['app'], ['firewall']],
  'return code': '200',
  'return info': 'Advanced path rule has been delivered successfully'
}
```

Figure 5: DDQN-Based Intelligent Path Generation

When the traffic passes through the security function node of the MEO satellite, the system first classifies the traffic based on the IP address and port number, and determines whether each flow is malicious or not. For a single detection module, its detection performance is evaluated by metrics such as accuracy, precision, recall, F1-score, and malicious flow recognition capability. However, for the same network flow, different detection modules may arrive at different judgment results, and the detection conclusions of each module in the security SFC may be inconsistent. Therefore, it is difficult to uniformly measure the performance of the overall security SFC in malicious traffic detection by relying only on traditional statistical metrics. To address this issue, in this experiment, we formulate a performance determination rule for the intelligently orchestrated Security SFC path: once any of the detection modules in the path determines a certain traffic as malicious, the traffic is regarded as malicious by the overall SFC. Further, this paper adopts the result of the highest percentage of the traffic being judged as a certain type in each detection module as the final categorization output of the security SFC. Figure 6 compares the malicious traffic detection performance of the baseline preset path with that of the path orchestrated using the DDQN algorithm across several evaluation metrics.

As shown in Figure 6(a), the accuracy of the DDQN-orchestrated path is higher than that of the preset path. This improvement stems from the DDQN model's ability to selectively include security function modules corresponding to the actual traffic characteristics.

Figure 6(b) presents the F1 score, which reflects the overall detection effectiveness by considering both precision and recall. The F1 score of the DDQN-based path improves by approximately 3.1% compared to the baseline path, indicating enhanced overall detection capability.

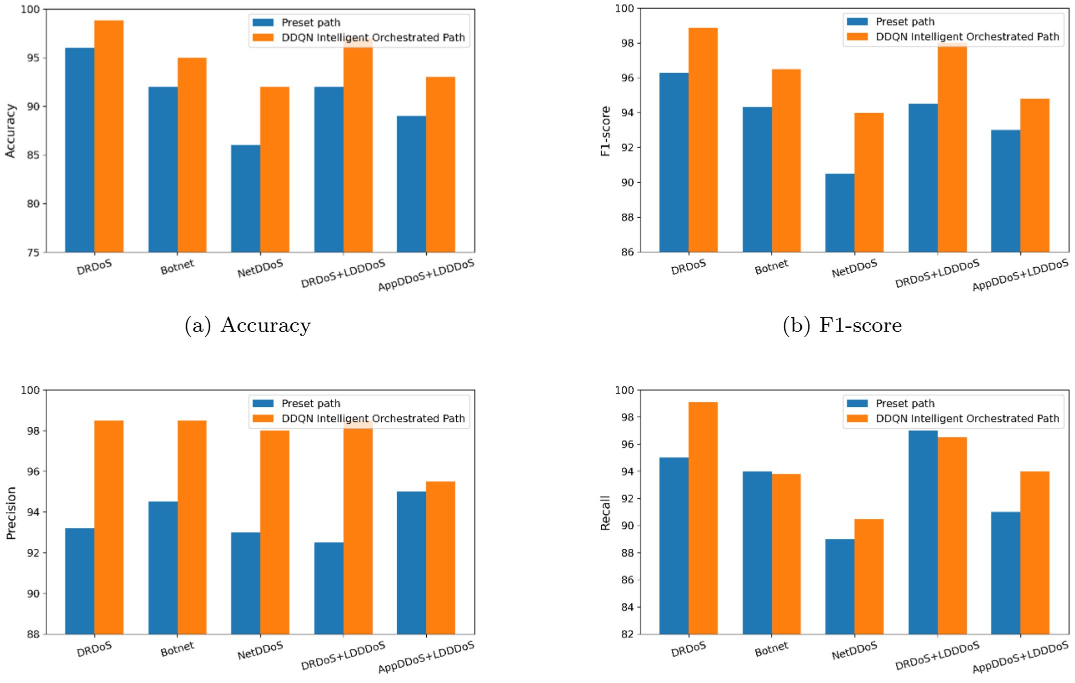
Figure 6(c) shows the precision metric, which captures the proportion of correctly identified malicious traffic among all traffic flagged as malicious. The precision of the DDQN-orchestrated path reaches 96.9%, compared to 92.6% in the baseline path, representing a gain of 4.4%.

Finally, Figure 6(d) illustrates the recall, measuring the ability of the system to detect anomalous traffic. In most attack categories, the DDQN-based path achieves higher or comparable recall values relative to the preset path, demonstrating improved or sustained detection coverage.

Figure 7 compares the average delay between the predefined path and the path orchestrated by the DDQN-based intelligent scheduling mechanism under both attack and normal traffic conditions.

When normal traffic passes through the satellite network, the preset path requires it to sequentially traverse multiple security function nodes on the MEO satellites, leading to higher transmission delays. In contrast, under DDQN orchestration, normal traffic can be transmitted to the LEO satellites by passing through only firewall nodes, which significantly reduces path complexity and latency.

When the input consists of mixed normal and DDoS attack traffic, the DDQN scheduling policy prioritizes forwarding normal traffic along a primary path containing fewer detection modules, thereby reducing processing delays. Identified malicious traffic is directed to specific



(a) Accuracy

(b) F1-score

(c) Precision

(d) Recall

Figure 6: Detection Performance

firewall nodes for interception.

Overall, the DDQN orchestration mechanism effectively reduces average delay by avoiding redundant detection operations and excessive forwarding, while still ensuring security. As shown in Figure 7, the average delay of the preset path is approximately 59 seconds,whereas the DDQN-optimized path reduces the delay to about 30 seconds, which represents an improvement of approximately 49.95%. This demonstrates the DDQN algorithm's significant advantages in path optimization and delay reduction.

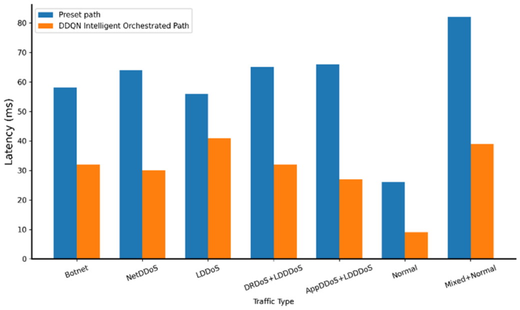


Figure 7: Traffic Transmission Delay

Figure 8 illustrates the total load generated by different traffic types transmitted to the satellite network when using the predefined paths compared with the paths obtained through DDQN-based intelligent scheduling.

Since the SFC load is primarily due to the forwarding and processing overhead introduced by detection modules along the path, we use the sum of the processing rates of all detection modules in each path as an approximate measure of the total load. This value is normalized to facilitate analysis and comparison.

At the beginning of the experiment (i.e., before time T_1), the system receives LDDoS attack traffic. In the subsequent stages (T_1 to T_4), the network sequentially receives:

- network-layer DDoS traffic,
- hybrid DRDoS and LDDoS traffic,
- hybrid application-layer DDoS and LDDoS traffic,
- normal traffic.

The intelligent scheduling mechanism dynamically adjusts the service paths to avoid unnecessary processing overhead, thereby reducing system load during periods of normal traffic and adapting accordingly under various attack scenarios.

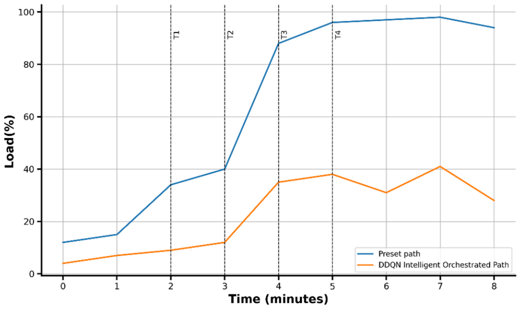


Figure 8: Service Path Load

In the preset path configuration, all incoming traffic must pass through all detection modules in sequence, resulting in a significant increase in SFC load during the high-intensity attack phase (T_2 to T_4). With the DDQN scheduling mechanism, the system is able to dynamically select a more lightly loaded path, containing fewer detection modules and effectively avoiding unnecessary processing, thereby maintaining a lower and more stable load level across all phases. Especially during high-load periods, the path selected by DDQN can significantly alleviate the resource utilization caused by multi-stage detection processes, demonstrating its advantage in adaptive scheduling when coping with dynamic attack traffic.

5 Conclusion and Future Work

This paper focuses on the STIN scenario and builds a security service framework with knowledge-aware perception and intelligent orchestration capabilities based on a three-layer SDN-enabled STIN architecture. The proposed system achieves real-time perception and synchronization of traffic characteristics, detection capabilities, and node resource states. By leveraging a DDQN to analyze multidimensional information within the knowledge base, the system dynamically optimizes the orchestration strategy of security function nodes and generates customized service paths for different attack scenarios. Experimental results demonstrate that the proposed method can be effectively applied in STIN environments, achieving a high attack detection rate while significantly reducing the load on functional nodes and the end-to-end transmission latency. Future research will further extend this work in three directions: (1) algorithm optimization and distributed deployment in large-scale environments, (2) integration of practical constraints in space–ground communications, and (3) resource scheduling through the combination of reinforcement learning and adaptive control. These efforts aim to enhance both the applicability and depth of the proposed method, providing a solid theoretical and engineering foundation for constructing an intelligent, secure, and controllable space–terrestrial integrated network.

6 Acknowledgments

- This paper is supported by NSFC under Grant No. 62341102, and the National Key R&D Program of China under Grant No. 2018YFA0701604.

References

[1] Statista. Global mobile network coverage and internet access statistics as of july 2023. [Online], 2023. https://www.statista.com/statistics/.

[2] 3GPP. 3gpp ts 33.501: Security architecture and procedures for 5g system, release 17. Standard, 3rd Generation Partnership Project (3GPP), Sophia Antipolis, 2022.

[3] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato. Space-air-ground integrated network: A survey. *IEEE Communications Surveys & Tutorials*, 20(4):2714–2741, 2018.

[4] J. Li, X. Li, C. Li, C. Wang, and J. He. A review of leo satellite network security research. In *Proceedings of the 2nd International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI)*, pages 496–501, Zakopane, Poland, 2023. IEEE.

[5] D. Bhamare, R. Jain, M. Samaka, et al. A survey on service function chaining. *Journal of Network and Computer Applications*, 75:138–155, 2016.

[6] C. Jiang, X. Wang, J. Wang, et al. Security in space information networks. *IEEE Communications Magazine*, 53(8):82–88, 2015.

[7] W. Guo, J. Xu, Y. Pei, et al. A distributed collaborative entrance defense framework against ddos attacks on satellite internet. *IEEE Internet of Things Journal*, 9(17):15497–15510, 2022.

[8] S. Deng, M. Li, and H. Zhou. Dynamic security sfc branching path selection using deep reinforcement learning. *Intelligent Automation & Soft Computing*, 37(3), 2023.

[9] G. Li, H. Zhou, B. Feng, et al. Horizontal-based orchestration for multi-domain sfc in sdn/nfv-enabled satellite/terrestrial networks. *China Communications*, 15(05):77–91, 2018.

[10] T. Li, H. Zhou, H. Luo, and S. Yu. Service: A software defined framework for integrated space-terrestrial satellite communication. *IEEE Transactions on Mobile Computing*, 17(3):703–716, 2018.