Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.084537

REVIEW

A Survey on AI-Integrated Detection Technologies for Intelligent Communication Networks

Jingfu Yan, Huachun Zhou*, Xiaojing Fan and Aoran Huang

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China

*Corresponding Author: Huachun Zhou. Email: hchzhou@bjtu.edu.cn

Received: 24 April 2026; Accepted: 02 July 2026; Published: 28 August 2026

ABSTRACT: Artificial Intelligence (AI) has been widely used to detect complex attacks and abnormal behaviors in intelligent communication networks. However, existing studies are often limited to a single scenario or technical route, making it difficult to systematically explain the roles, boundaries, and collaboration mechanisms of different AI-integrated detection technologies. To address this gap, this paper reviews AI-integrated detection technologies in intelligent communication networks from both scenario and technical perspectives. From the perspective of application scenarios, the paper summarizes the requirements and characteristics of AI detection in cloud networks, edge computing, satellite and space networks, and Internet of Things (IoT) environments. From the perspective of technical implementation, the paper first reviews AI detection methods based on programmable data planes (PDPs). These methods are analyzed in terms of in-network feature extraction, lightweight inference, temporal behavior modeling, and data-plane capability boundaries. The paper then introduces knowledge-driven detection methods and explains how security knowledge modeling enhances interpretability and supports attack analysis. Next, blockchain-enabled AI detection is reviewed, focusing on distributed trust, privacy-preserving collaboration, trusted mechanisms, and auditability of detection results. Finally, the paper discusses recent studies on Large Language Models (LLMs) for security detection, highlighting their potential in contextual modeling, reasoning-based detection, explanation generation, and decision support, as well as their deployment risks. Through a comparative analysis of different technical routes, this paper clarifies that these technologies should be deployed in a layered collaborative architecture rather than simply combined at the same execution layer. Specifically, programmable switches (PSs) are more suitable for real-time sensing and preliminary filtering, knowledge-driven methods for semantic correlation and attack reasoning, blockchain (BC) for trusted evidence management and audit support, and LLMs for high-level explanation and decision support. Finally, key challenges and future directions are discussed.

KEYWORDS: AI; traffic detection; programmable switches; knowledge-driven; blockchain; large language models

1 Introduction

Artificial Intelligence (AI) technologies have alleviated the limitations of traditional communication network security, which relies heavily on manual rule configuration and expert-driven modeling [1,2]. AI improves the detection capability of intelligent communication networks. It also provides new approaches for threat detection and defense in complex network environments [3,4].

In intelligent communication networks, the core of AI-based detection lies in data-driven modeling of communication behaviors. By modeling packet-level, flow-level, and behavior-level features, AI-based detection methods can extract discriminative representations from high-dimensional and noisy observation

data. These features can then be used to identify abnormal behaviors [5,6]. In addition, AI methods capture temporal correlations and construct behavioral fingerprints. This enhances the detection of slow and stealthy attacks [7].

In dynamic and distributed intelligent communication networks, multi-node collaborative AI detection has become an important way to improve overall detection performance [8,9]. In this context, AI detection has become a key component of the security system. By analyzing long-term communication behaviors and detection results, AI models support security assessment, risk classification, and defense strategy adjustment [10,11]. Detection outputs are involved in decision-making processes such as strategy generation and resource scheduling. This promotes the evolution of security mechanisms from static configuration to dynamic coordination.

However, AI detection in intelligent communication networks still faces limitations in real-time performance, scalability, interpretability, reliability, and model transferability [6,12–14]. Accordingly, related studies have gradually shifted from optimizing single-model performance to reconstructing the overall detection system. To improve real-time processing in high-speed traffic scenarios, programmable switches (PSs) and data-plane computing capabilities have been used to move sensing and preliminary analysis into the network, thereby shortening the detection chain. To enhance interpretability, knowledge graphs (KGs) have been introduced to semantically organize detection outputs and support association reasoning, improving the consistency between detection results and attack behaviors [15]. Meanwhile, blockchain (BC) technology has been used to provide tamper-resistant evidence storage and audit mechanisms for detection processes and results, helping address trust deficiency and responsibility attribution in cross-domain collaborative environments [16]. On this basis, Large Language Model (LLM)-based detection methods have begun to attract attention because of their potential in contextual understanding, implicit feature mining, and reasoning-assisted security analysis [14].

AI detection has become a fundamental capability in intelligent communication network security. However, a single technical perspective cannot fully address complex security requirements. Therefore, it is necessary to systematically review detection methods from multiple dimensions, including execution mechanisms, semantic modeling, trust management, and unified reasoning. Based on this, this paper reviews AI detection methods based on programmable data planes (PDPs), knowledge-driven detection models, BC-enabled distributed detection mechanisms, and large model-based approaches.

1.1 Background and Motivation

AI-driven traffic detection and anomaly identification have become important techniques in network security [17]. Existing studies mainly focus on feature learning and model inference in centralized or control-plane settings. Their effectiveness often relies on relatively stable traffic distributions and sufficient computational resources [18]. However, in real-world networks, traffic scale and service patterns change continuously. High-speed data forwarding introduces additional delays in feature collection, centralized analysis, and policy deployment. This makes it difficult to achieve fast detection and response to abnormal behaviors.

Existing AI detection methods lack the ability to abstract features into knowledge. As a result, detection results are difficult to associate with specific attack behaviors. This leads to limitations in interpretability [19,20]. When attack strategies change or detection scenarios shift, detection models often need to be retrained. Their outputs are also difficult to trace back to clear causal relationships. This limits their transferability in security scenarios [21]. In addition, existing methods still lack a systematic knowledge-based representation of multi-stage attack chains, cross-entity associations, and long-term behavioral evolution.

This makes it difficult to support the understanding of complex and persistent attacks [14,22]. From the perspective of communication security, cross-domain data collection exposes detection data and intermediate features to privacy leakage and misuse risks during collection, transmission, and sharing [23–25].

When detection models need to share results and model parameters, these results and parameters may be tampered with in distributed environments, thereby weakening the reliability of collaborative detection [26,27] At the same time, detection processes, model versions, and decision contexts often lack audit mechanisms. This makes post-event reproduction and review difficult. This problem is especially prominent in cross-domain collaborative detection and long-term operating systems [28].

As attack behaviors continue to evolve, detection scenarios become more diverse, and semantic associations become more complex. These changes place higher demands on the generalization ability and contextual understanding of detection models. Detection methods based on feature engineering and supervised learning rely heavily on high-quality labeled data and existing attack distributions. As a result, they struggle to handle zero-day attacks and unknown threats effectively [28,29]. Therefore, it is necessary to model the contextual associations, semantic structures, and evolution logic of attack behaviors. This can help detection methods better respond to new types of attacks.

Based on the above analysis, existing AI-driven network security detection studies have formed several new technical routes at different levels. However, most studies are still designed from a single perspective. They usually focus on specific operating conditions or local security requirements. Overall, these methods have developed in parallel, but they lack systematic integration. Therefore, it is necessary to conduct a systematic review of existing studies.

1.2 Scope and Objectives

Based on the above background, security detection in intelligent communication networks is facing several challenges. Detection objects are becoming more complex. Technical systems are becoming more diverse. Security requirements are also increasing. To address these challenges, this paper covers several typical scenarios, including cloud computing, edge computing, satellite networks, and the Internet of Things. It focuses on the bottlenecks and frontier solutions of AI detection in real-time performance, interpretability, reliability, auditability, and advanced semantic understanding. It also explores four research directions: lightweight detection based on PSs, knowledge-driven interpretable detection, BC-enabled distributed trusted detection, and large-model-enabled semantic detection.

Since existing studies differ in datasets, feature forms, evaluation metrics, and experimental environments, this paper does not directly generalize the reported performance gains of different methods. Instead, it analyzes their functional roles, operating mechanisms, applicable conditions, and collaborative potential. Specifically, this paper focuses on the following four questions:

Question 1: How can low-latency and real-time detection be achieved in high-speed networks?

Question 2: How can the interpretability of detection results and the understanding of attack semantics be improved?

Question 3: How can trusted collaboration, privacy protection, and result auditability be ensured in distributed detection processes?

Question 4: How can semantic modeling, contextual understanding, and reasoning capabilities for complex unknown attacks be enhanced?

This paper illustrates the relationships among the four technical directions from a system-level perspective, as shown in Fig. 1. Programmable switches are mainly responsible for feature acquisition, rule configuration, and real-time detection. They upload detection features, alarm information, and behavior records to the knowledge-driven module and the BC module. The knowledge-driven module is used to interpret detection results, correlate attack stages, and conduct traceability analysis, while storing the knowledge correlation results on the BC. The BC provides trusted support for detection evidence, user behavior records, and cross-domain knowledge sharing. Large models integrate detection results, knowledge correlations, and trusted evidence at the upper layer, further enabling knowledge correlation mining and explanation generation. In this way, the four technical directions form a collaborative detection framework consisting of bottom-layer real-time detection, middle-layer knowledge analysis and trusted evidence storage, and upper-layer semantic reasoning [14,30,31].

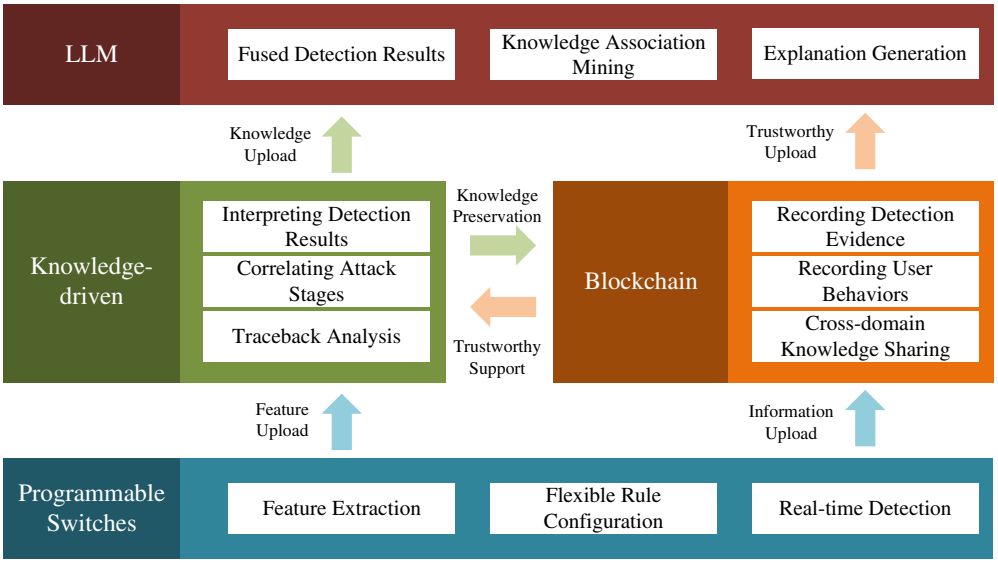

Figure 1: Framework for AI-enhanced detection in intelligent communication networks.

1.3 Comparison with Existing Surveys

Existing surveys have discussed AI-based security detection from different perspectives, including programmable-switch-based detection [32,33], knowledge-driven detection [34], BC-enabled trusted security frameworks [35,36], and LLM-based detection paradigms [14,37]. These studies provide useful foundations for understanding the development of intelligent security detection. However, their review focuses, covered scenarios, and involved technologies are different. Some studies focus on a single technical direction, while others discuss the integration of multiple technologies in specific network or security contexts. To this end, this paper summarizes representative surveys related to PDPs, knowledge-driven security analysis, BC-enabled trusted detection, and LLM-based cybersecurity. These surveys are compared to clarify whether they cover scenario-specific detection requirements and whether they analyze cross-technology collaboration among PS, KG, BC, and LLMs, as shown in Table 1.

Surveys on PSs mainly focus on line-rate detection, in-network processing, programmable data-plane functions, and PDP-based security applications. These studies show the potential of moving security functions closer to the data plane, especially for high-speed traffic monitoring and line-rate defense [38]. However, they usually emphasize lightweight deployment, packet- or flow-level processing, and traffic-level detection. Their discussion of knowledge reasoning, trusted verification, and semantic understanding is still limited [32,33,38].

Table 1: Representative surveys related to AI-integrated security detection.

Reference	Year	Main Focus	Scenarios	Technologies	Limitations
[32]	2021	P4 programmable data-plane switches, including taxonomy, applications, challenges, and future trends.	Not scenario-specific	PS	Provides a broad survey of programmable data-plane switches, but AI-based security detection is not the central review thread.
[33]	2023	Data-plane programming with P4, including fundamentals, advances, and applied research.	Not scenario-specific	PS	Focuses on P4 programming models and applications; security detection and cross-technology integration are only partially covered.
[38]	2023	Programmable-switch-enabled line-rate security detection and defense.	Not scenario-specific	PS	Focuses on in-network security mechanisms; cross-technology integration is limited.
[19]	2024	Cybersecurity KG construction, including knowledge extraction, representation, fusion, storage, reasoning, and application.	Not scenario-specific	KG	Mainly focuses on KG construction technologies, while the connection with real-time detection, distributed trusted collaboration, and multi-technology integration remains limited.
[39]	2023	IoT IDS mechanisms, deployment strategies, datasets, and future directions.	IoT	KG	Knowledge is mainly reflected in signature-/knowledge-based IDS; explicit KG-based semantic reasoning is not the core focus.
[35]	2023	BC and AI for secure 6G communications.	6G wireless and space-air-ground integrated networks	BC	Provides a networking-oriented view; detection is not the core review thread.
[36]	2024	BC- and federated-learning-based intrusion detection for edge-enabled industrial IoT networks.	Edge-enabled industrial IoT	BC	Focuses on BC and federated learning for IIoT intrusion detection, while PS-, KG-, and LLM-based detection mechanisms are not systematically compared.

(Continued)

Table 1 (continued)

Reference	Year	Main Focus	Scenarios	Technologies	Limitations
[14]	2025	LLM applications in cybersecurity.	Not scenario-specific	LLMs	Provides a broad cybersecurity view; detection is not independently systematized.
[40]	2025	LLM-based, graph-based, and explainable AI methods for NIDS.	Not scenario-specific	KG, LLMs	Scenario coverage and deployment constraints remain limited.
[37]	2025	Cybersecurity-oriented LLM construction and applications.	Not scenario-specific	KG, LLMs	Detection processes and deployment boundaries are not sufficiently separated.

Note: LLMs = large language models; IDS = intrusion detection system; NIDS = network intrusion detection system. "No" indicates that the corresponding item is not explicitly described.

Knowledge-driven surveys mainly focus on cybersecurity KGs, knowledge representation, knowledge reasoning, and KG construction. These studies show that knowledge can support security analysis, threat intelligence correlation, situational awareness, and the interpretability of detection results. Some Internet of Things (IoT) IDS surveys also discuss knowledge-based detection, but they mainly regard it as signature- or rule-based detection rather than explicit KG-based semantic reasoning [39]. Therefore, many existing studies still focus on knowledge construction, KG applications, or network management [19]. The role of knowledge in the complete detection process has not been fully summarized. In particular, the links among knowledge acquisition, reasoning, detection decision-making, result interpretation, and response are still not clear enough [19].

BC-related surveys mainly discuss trusted networking, privacy protection, distributed collaboration, and secure communication systems. They cover scenarios such as communication networks, cloud-edge networks, 6G wireless networks, space-air-ground integrated networks, and edge-enabled industrial IoT networks [31,35,36]. These studies show the value of BC in improving data trust, process traceability, privacy protection, and collaborative security [31,35]. However, security detection is often treated as one application of BC-based network security or BC-AI integration. The integration of BC with PDPs, knowledge-driven reasoning, and LLM-based detection remains insufficient [35,36].

LLM-related surveys provide a broad view of large-model-based cybersecurity. They cover threat intelligence, vulnerability detection, malware detection, anomaly detection, security agents, and cybersecurity-oriented LLM construction. Some studies also discuss the integration of LLMs with graph-based methods, explainable AI, or knowledge-driven mechanisms for network intrusion detection [40]. However, detection tasks are often discussed together with other security applications, such as malware analysis, phishing prevention, vulnerability analysis, secure code generation, and security operation support [37]. The internal logic of detection methods is not sufficiently separated. In addition, real-time detection, data-plane deployability, and system-level implementation constraints are still not fully analyzed [14,40].

Overall, existing surveys provide useful foundations for AI-based security detection, but they remain fragmented in terms of technical focus, scenario coverage, and detection-process analysis [31,38]. Few studies provide a unified review of AI-integrated security detection across PS, KG, BC, and LLMs. In addition, cross-scenario comparison is still limited, especially for heterogeneous intelligent communication environments such as cloud computing, edge computing, end devices, IoT, and space-air-ground integrated networks [35,36,39]. Therefore, this paper further reviews AI-integrated security detection in intelligent

communication networks from a multi-technology and cross-scenario perspective. It focuses on lightweight detection supported by PSs, knowledge-driven explainable detection, BC-enabled trustworthy detection, and LLM-enabled semantic detection.

1.4 Main Contributions

Compared with existing surveys, this paper reviews AI-integrated detection technologies from the perspective of the detection chain, where sensing, semantic correlation, trust verification, and high-level reasoning are treated as different functional layers. The main contributions of this paper are summarized as follows.

- **A detection-chain-oriented taxonomy is developed.** Existing surveys usually analyze PSs, KGs, BC, and LLMs separately. This paper reorganizes these technologies according to their roles in the detection chain, including low-latency traffic sensing, semantic knowledge correlation, trusted evidence management, and explanation-oriented decision support.
- **Capability boundaries and engineering constraints are compared across four technical routes.** Instead of only summarizing reported detection performance, this paper analyzes the deployment locations, data granularity, real-time capability, resource overhead, interpretability, trustworthiness, and practical limitations of programmable-switch-based detection, knowledge-driven detection, BC-enabled detection, and LLM-based detection.
- **Scenario-specific detection requirements are linked with technical applicability.** This paper compares cloud computing, edge computing, satellite networks, and IoT environments in terms of traffic scale, latency requirement, resource constraint, data distribution, and collaboration mode. Based on these differences, it discusses which technologies are more suitable as core detection components and which technologies are more suitable as auxiliary support mechanisms in each scenario.
- **A layered collaboration view is proposed for multi-technology joint detection.** This paper clarifies that the four technologies should not be simply combined at the same execution layer. Programmable switches are more suitable for real-time sensing and preliminary filtering, knowledge-driven methods for semantic correlation and attack reasoning, BC for trusted evidence management and audit support, and LLMs for high-level explanation and decision support.

1.5 Organization of this Paper

The remainder of this paper is organized according to a scenario–technology–integration logic, as shown in Fig. 2. Section 2 analyzes the differentiated detection requirements of cloud computing, edge computing, satellite networks, and IoT environments. Sections 3–6 review four technical routes, including programmable-switch-based detection, knowledge-driven detection, BC-enabled detection, and LLM-based detection. For each route, this paper focuses on functional roles, representative methods, deployment boundaries, and interfaces with other technologies. Section 7 summarizes technical challenges, scenario-specific unresolved problems, and future research directions.

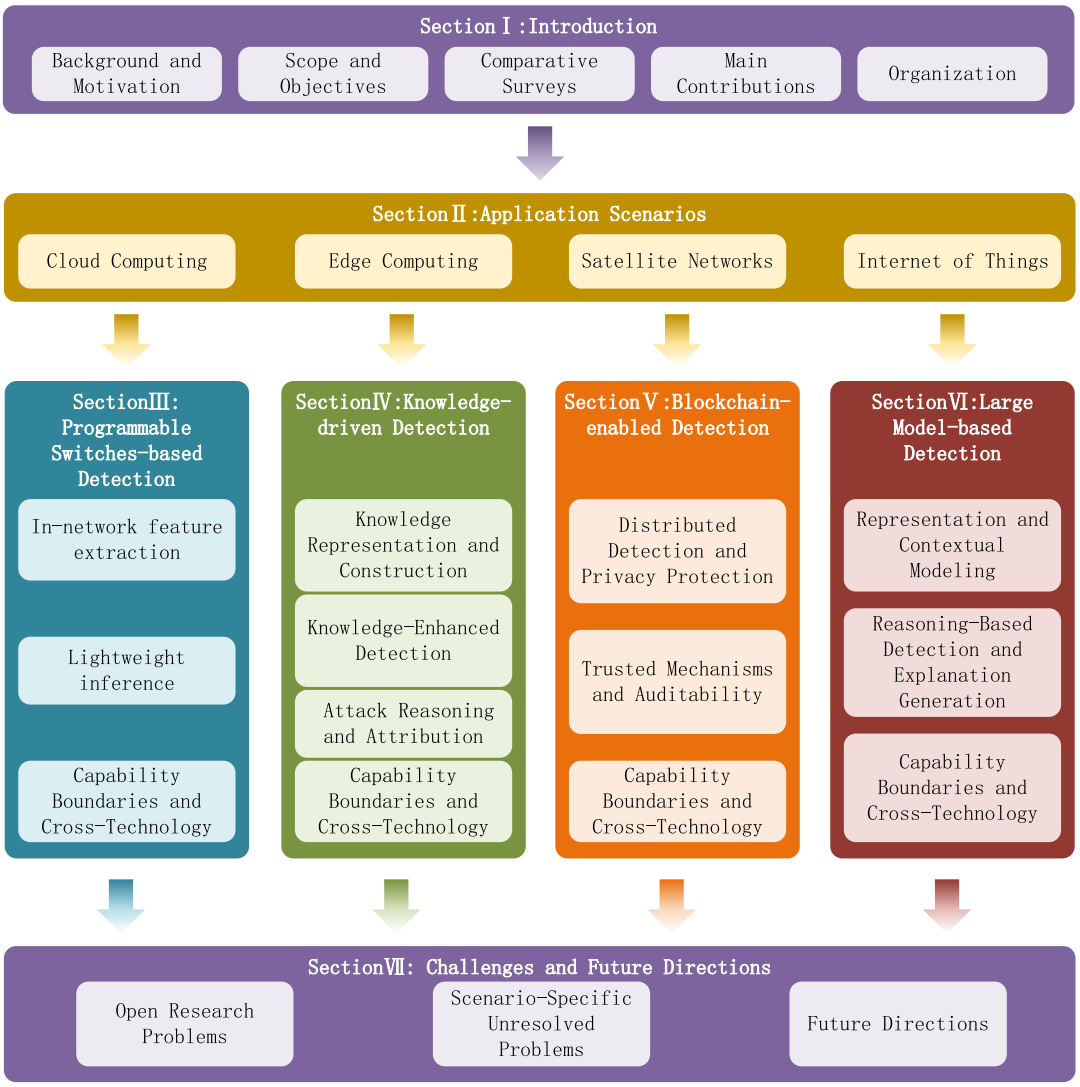

Figure 2: Outline of the paper.

2 AI-Based Detection in Different Scenarios

The operating environment of intelligent communication networks is highly heterogeneous, dynamically evolving, and cross-domain collaborative. Typical scenarios, including cloud computing, edge computing, satellite networks, and the IoT, differ significantly in network scale, resource conditions, data distribution, and security requirements [41–47]. Therefore, AI-based detection methods should be adapted to scenario-specific constraints rather than being designed only as general-purpose models.

In cloud computing environments, large-scale distributed architectures, elastic resource scheduling, multi-tenant coexistence, and dynamic service workloads make abnormal behaviors diverse and difficult to define in advance [41,48]. Cloud platforms usually provide relatively sufficient computing resources and centralized data collection capabilities, so AI-based detection mainly emphasizes high detection accuracy, scalable analysis, and robustness to workload fluctuations. Supervised learning is effective for abnormal behaviors with clear security semantics, such as known attack traffic, abnormal logins, and unauthorized access, but it is less effective for unknown attacks because it depends on existing labels and known attack

distributions [49–51]. In contrast, unsupervised anomaly detection can discover gradual deviations in resource usage, access patterns, or user behaviors without relying on labeled data, but detection based only on statistical deviation may still lead to false alarms during elastic scaling or workload peaks [49,52]. Thus, cloud-oriented AI detection should combine high-accuracy classification with scalable anomaly discovery.

In edge computing and cloud-edge collaborative scenarios, security detection faces stricter latency and resource constraints. If all security-related data are uploaded to the cloud for analysis, the detection and response process may be affected by network latency and bandwidth conditions [53]. Deploying AI models at the edge allows data to be analyzed near its source, which shortens the security awareness and response chain [54]. However, edge nodes usually have limited computing power, storage capacity, and energy supply. Therefore, edge-side detection models need to be lightweight [55]. Edge scenarios also involve privacy-sensitive data, and local processing can reduce the exposure risk caused by centralized transmission and cloud storage [56–58]. When network connectivity is unstable, edge-side AI can maintain basic security awareness and preliminary decision-making capability without fully relying on the cloud [59]. In addition, edge security protection often requires dynamic adjustment among latency, detection accuracy, computing resource usage, and energy consumption. Therefore, AI-based detection in edge scenarios should emphasize low latency, lightweight inference, local detection, and privacy preservation.

Satellite and space networks have stronger dynamics and harsher deployment constraints. Space information networks are usually composed of low Earth orbit/geostationary Earth orbit (LEO/GEO) constellations, inter-satellite links, ground gateways, and terrestrial networks, making their structure more complex than traditional satellite–ground links [42]. In LEO constellations, high-speed satellite movement causes rapid topology changes and frequent link handovers, while long propagation delay, intermittent connectivity, and limited onboard computing resources make centralized real-time detection difficult [42]. Existing machine learning (ML)- and deep learning (DL)-based satellite intrusion detection studies still face limitations such as data scarcity, class imbalance, and dependence on oversampling [46,60]. For satellite–ground collaborative scenarios, deeper intrusion detection can be performed on the ground, while trusted mechanisms can improve system security and reliability [61]. Therefore, AI-based detection in satellite networks should focus on hierarchical deployment, dynamic adaptation, and trusted space–ground collaboration.

IoT environments are characterized by massive devices, strong heterogeneity, scarce labels, noisy traffic, and limited terminal resources [39,62]. The wide deployment and diversity of IoT devices make data integrity, availability, and confidentiality difficult to guarantee simultaneously [63]. Since many terminal devices have limited computing power, storage capacity, and energy supply, heavyweight centralized security mechanisms are difficult to apply directly [64]. AI-based IDSs can learn normal behavior patterns from historical and real-time data, thereby improving the detection of unknown and variant attacks [65]. However, IoT attacks show cross-layer and cumulative characteristics, including malware, distributed denial-of-service (DDoS) attacks, data tampering, privacy leakage, and poisoning attacks [66,67]. Although ensemble ML and DL methods can improve detection performance, their computational cost may introduce new risks in resource-constrained or latency-sensitive IoT scenarios [68]. Therefore, AI-based IoT detection should emphasize lightweight models, distributed deployment, and hierarchical collaboration among terminals, edge nodes, and cloud platforms.

As shown in Table 2, the four scenarios impose different detection requirements. Cloud computing requires high-accuracy and scalable detection for large-scale and dynamic resources. Edge computing requires lightweight and local detection under latency, resource, and privacy constraints. Satellite networks require hierarchical detection and trusted collaboration due to dynamic topology, unstable links, and limited onboard resources. IoT requires lightweight and distributed detection for heterogeneous devices,

scarce labels, and constrained terminals. These differences indicate that AI-based IDSs are evolving from general-purpose models toward scenario-adaptive and collaborative detection frameworks [39,42].

Table 2: Security requirements and AI detection capabilities across different communication scenarios.

Dimension	Cloud Computing	Edge Computing	Satellite Networks	IoT
Scenario Characteristics	Large-scale, multi-tenant, and dynamically changing resources	Low latency and resource constraints	Dynamic topology and unstable links [42]	Device heterogeneity, scarce labels, and limited terminal resources [63,64]
Main Detection Requirements	High accuracy, scalability, and robustness to workload fluctuation [49]	Lightweight and local detection	Hierarchical detection and trusted collaboration [69]	Lightweight models and distributed detection [68]
Data Characteristics	Multi-source, large-scale, heterogeneous, dynamic, and partially labeled data [70]	Distributed and privacy-sensitive data [56]	Scarce and imbalanced data [71]	Noisy, heterogeneous, cross-domain, and device-dependent data [72]
Major Threats	Known attack traffic, behavioral anomalies, unauthorized access, and resource abuse	Latency-sensitive attacks and data leakage [56]	Link attacks, interference, malicious nodes, and spoofing [73]	Malware, DDoS, data tampering, privacy leakage, and cross-layer attacks [66,67]
Typical Detection Paradigms	Supervised learning, unsupervised anomaly detection, and scalable cloud-side analysis [52,74]	Edge intelligence, federated learning, and RL [54]	Distributed learning, trust modeling, and hierarchical detection [75]	Lightweight ML/DL, hierarchical IDS, and distributed anomaly detection [65]
Deployment Locations	Cloud center and security operation platform	Edge nodes and cloud–edge platforms	Onboard satellites and ground stations	Terminals, gateways, edge nodes, and cloud platforms
Representative Supporting Technologies	KG, BC & LLMs	PS & BC	BC [76] & KG [77]	PS, BC & LLMs

3 AI Detection Methods Based on Programmable Switches

As PSs gradually acquire limited but efficient in-network computing capabilities, part of the AI-based detection process has been moved into the data plane for low-latency malicious traffic and anomaly detection [30,38,78]. Unlike surveys that classify related studies mainly by learning models, this section organizes programmable-switch-based AI detection by functional role. Specifically, PSs support feature extraction [79–81], approximate traffic measurement [82,83], lightweight inference [84,85], and temporal state observation [7,86]. These capability boundaries determine which tasks can be executed in the forwarding path and which should be transferred to the control plane or upper-layer security modules.

3.1 In-Network Feature Extraction and Lightweight Detection

In-network feature extraction and lightweight detection aim to achieve fast security awareness in line-rate forwarding environments. These methods use PDPs to extract packet-level or flow-level features during forwarding and support rapid classification, compact traffic measurement, or preliminary anomaly screening, thereby reducing the delay caused by traffic export and control-plane analysis. However, complex detection models still usually require control-plane or external computing support. Therefore, as shown

in Table 3, this subsection summarizes representative studies along three functional routes: in-switch lightweight inference, Sketch-based compact measurement, and data-plane-assisted collaborative detection.

Table 3: Representative in-network feature extraction and detection methods in programmable switches.

Reference	Method	Functional Route	Deployment	Reported Results
[84]	Flowrest based on RF	Tree-based in-switch inference	DP	Classification rate = 99.4%; Macro F1 > 96%; Weighted F1 > 99%; sub-microsecond inference latency.
[85]	SwitchTree based on RF	Tree-based in-switch inference	DP	Precision = 0.9662; Recall = 0.9725; F1 = 0.9693; attack detection rate = 97.36%; packet loss rate ≈ 0.26%.
[87]	In-Forest-M based on ensemble learning	Distributed multi-task inference	Multiple DPs	Average CFA difference = 16.39%; CFA differences on Abilene, GEANT, and DialtelecomCz are 1.15%, 1.57%, and 1.52%, respectively.
[82]	Mix Sketch/Co-Mix Sketch	Sketch-based compact measurement	DP	Co-Mix WMRE ≈ 0.001; Mix WMRE ≈ 0.01.
[83]	SP-Sketch	Sketch-based persistent-flow detection	DP	Detection accuracy > 95%; ARE < 0.1; MSE < 2; RR = 98.7%.
[88]	NetNN based on DNN	Data-plane-assisted DL detection	DP + CP	Packet accuracy = 83%; flow accuracy = 99%.
[89]	Mousikav2 based on binary decision trees and knowledge distillation	Distillation-based in-network classification	DP + CP	Accuracy = 97.66%; DT accuracy = 94.39%; packet processing latency ≈ 660 ns.
[90]	Mousika based on binary decision trees and knowledge distillation	Distillation-based lightweight inference	DP + CP	Flow size prediction accuracy = 94.95%; DT accuracy = 92.23%; packet loss rate = 0% at 100 Gbps; classification latency = 466 ns.

Note: DP: Data Plane; CP: Control Plane; DNN: Deep Neural Network; CFA: Classification Accuracy; WMRE: Weighted Mean Relative Error; ARE: Average Relative Error; MSE: Mean Squared Error; RR: Recall Rate; TPR: True Positive Rate; FPR: False Positive Rate; PR-AUC: Precision-Recall Area Under Curve.

Feature construction in PSs is shaped by both data-plane constraints and line-rate processing advantages. Feature design usually emphasizes low dimensionality, incremental updates, and approximate computation [32]. Meanwhile, PSs can parse packet headers and update packet-level or flow-level statistics during forwarding, providing timely inputs for real-time detection. For example, Gray et al. used the P4 pipeline to extract network metadata at high speed for ML-based intrusion detection [79], while Mittal et al. implemented fast updates of flow-level statistical features for switch-speed traffic classification [80,81]. Therefore, feature construction is a key prerequisite for programmable-switch-based real-time detection.

Fig. 3 illustrates the workflow of lightweight model deployment on programmable switches. Packets enter the programmable data plane through the parser, where packet headers and flow-level metadata are extracted. The trained ML model is then decomposed into executable rules and mapped into the P4 program. During forwarding, the match-action pipeline performs feature matching and rule execution, and generates detection results before packets leave through the deparser. This process shows that programmable switches can support low-latency inference when the model can be translated into match-action logic. Therefore,

decision trees (DTs) and random forests (RFs) are practical choices for data-plane detection because their feature-splitting logic can be mapped into P4-executable conditional rules [85,91]. Some studies further use feature sharing, model pruning, and parallel inference to support multi-model or multi-task detection under limited data-plane resources [84,87].

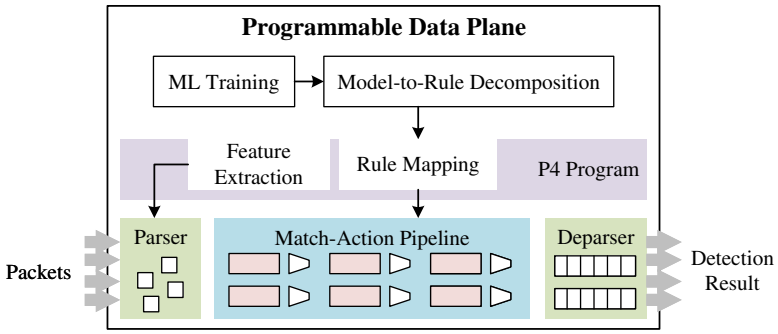

Figure 3: Lightweight model on PS.

Fig. 4 shows the implementation logic of Sketch-based compact measurement in programmable switches. The programmable data plane maps packet or flow identifiers into multiple buckets through hash functions and updates the corresponding counters in the match-action pipeline. $\mathcal{F}$ denotes the feature set, which generally consists of five-tuple information. This design significantly reduces memory consumption and allows the switch to maintain approximate traffic statistics at line rate. The generated compact measurements can be used to detect heavy flows or abnormal traffic frequency changes [92–94]. Therefore, approximate statistical structures, such as Sketch, are widely used in PSs to reduce storage and computational overhead. For example, Co-Mix Sketch further distributes these components across hierarchical nodes, reducing the resource pressure on a single switch [82]. SP-Sketch targets persistent flow detection by using a single Sketch structure and a probabilistic decrement mechanism to approximate sliding-window updates, thereby reducing memory consumption while maintaining high detection accuracy [83]. These studies indicate that Sketch-based measurement can provide compact traffic summaries for subsequent lightweight detection. They are suitable for front-end traffic measurement and anomaly screening. However, their outputs usually need to be further analyzed by control-plane or upper-layer detection modules.

Fig. 5 presents a data-plane-assisted deep learning detection framework. Different from Fig. 3, where lightweight rules are directly executed in the data plane, this framework separates feature collection and model inference across the programmable data plane and the control plane. Specifically, the data plane is responsible for parsing packets, collecting packet-level or flow-level features, and updating packet information during forwarding, while the control plane or external computing modules perform deeper analysis and execute deep model inference. The generated detection decisions are then decomposed into enforceable rules and redistributed to the programmable data plane, such as IP limitation rules or filtering policies. This architecture preserves the real-time sensing capability of programmable switches while avoiding the direct deployment of complex deep learning models in the forwarding pipeline. Due to the constraints of data-plane latency, storage, and computing resources, DL models are usually difficult to deploy completely inside switches. Therefore, data-plane/control-plane collaboration provides a practical compromise between detection accuracy and system overhead [88]. In addition, some studies compress the discriminative capability of deep models into lightweight decision structures through model distillation, enabling part of the model to be offloaded to the data plane for execution [90]. As a result, DL models are more suitable as

collaborative analysis modules in programmable-switch-based detection systems, rather than as strict line-rate real-time detectors in the data plane [89]. Although DL-based collaboration can enhance the ability to identify complex attacks, stealthy attacks and slowly evolving anomalies still require traffic observations over a longer time scale before they can be detected.

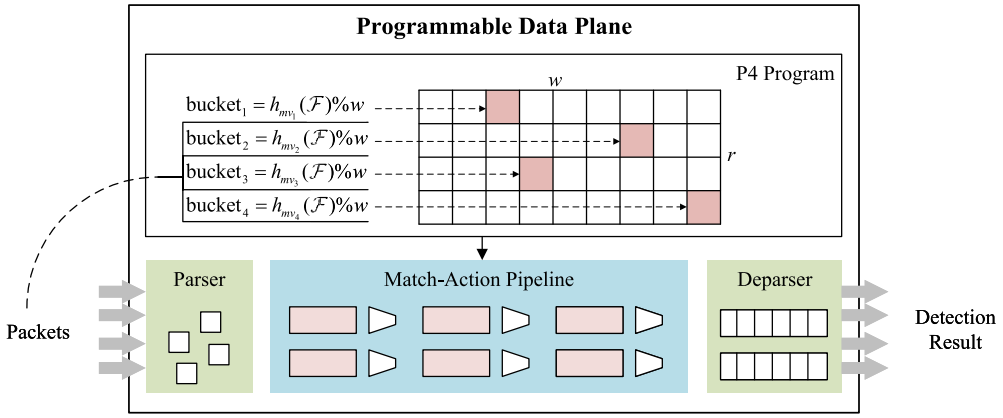

Figure 4: Sketch on PS.

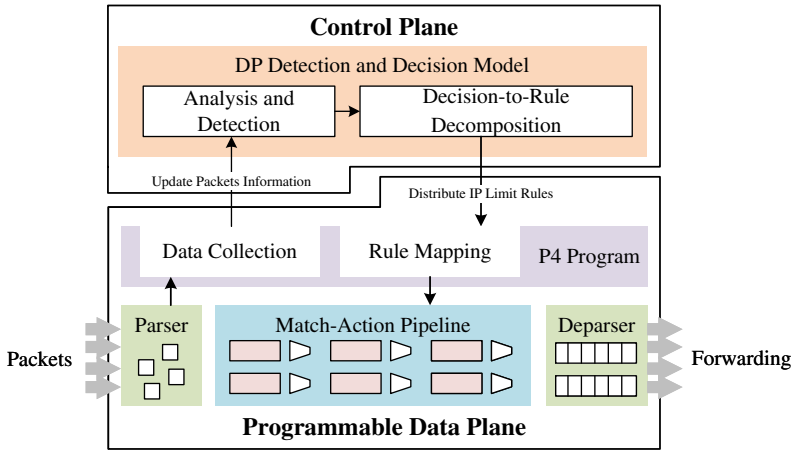

Figure 5: DL on PS.

The methods in Table 3 reflect different trade-offs. Tree-based methods, such as random forests and decision trees, are suitable for in-switch inference because their splitting rules can be mapped to match-action logic or conditional statements [84,85]. However, their model depth, feature number, and branching structure must be strictly controlled to satisfy pipeline constraints [84,87]. Sketch-based methods are more suitable for compact in-network measurement because they summarize flow size, persistence, or frequency under limited memory, but they mainly capture statistical deviations and cannot independently express rich attack semantics [83]. Distillation-based and data-plane/control-plane collaborative methods provide a compromise between model representation capability and data-plane deployability, but their training, updating, or complex inference processes still rely on the control plane or external computing modules [88–90]. Therefore, in-network feature extraction and lightweight detection are mainly suitable for front-end sensing, known attack screening, and high-throughput anomaly filtering. They remain limited in unknown

attack identification, cross-flow semantic correlation, and long-term attack evolution analysis, which require temporal modeling or upper-layer semantic analysis [7,95,96].

3.2 Temporal Behavior Modeling and Collaborative Detection

For traffic behaviors with temporal variation characteristics, PSs can continuously extract and update flow states in the data plane and construct temporal features or behavioral fingerprints [7,97]. This capability helps identify persistent attacks, slowly evolving anomalies, and multi-stage attack behaviors. Unlike lightweight detection based on short-term traffic features, temporal behavior modeling focuses on how traffic states evolve over time.

As Fig. 6 illustrated, a unified mechanism for temporal behavior modeling and collaborative detection on PSs. The programmable data plane first parses incoming packets and extracts temporal features from packet headers, timestamps, inter-packet intervals, and flow-state variations. These temporal features are processed by the match-action pipeline and the P4 program to support preliminary detection during packet forwarding. Compared with static feature extraction, this process emphasizes continuous state/window updates, allowing the data plane to capture short-term traffic changes and generate early detection results at line rate.

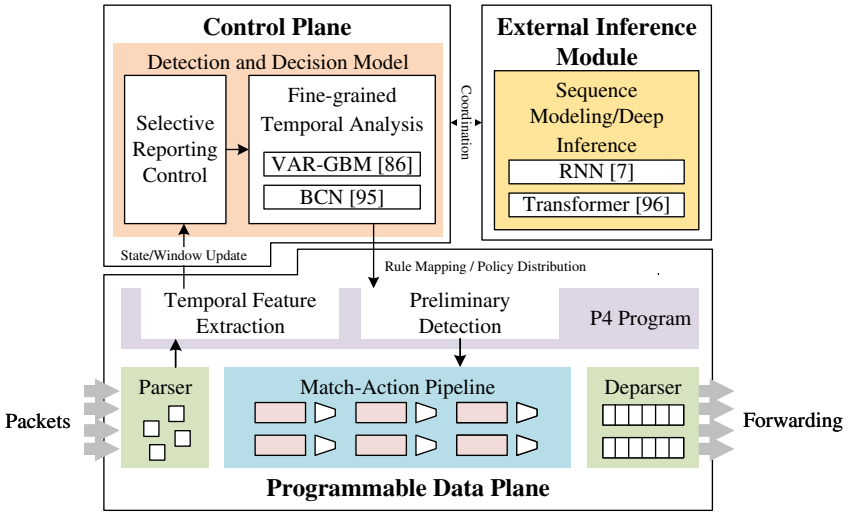

Figure 6: Temporal behavior modeling on PS [7,86,95,96].

Above the data plane, the control plane performs detection decision-making and reporting control. The selective reporting module determines which temporal features, suspicious flows, or low-confidence results should be uploaded, thereby reducing unnecessary communication and computation overhead. The fine-grained temporal analysis module can further analyze reported information using models such as VAR-GBM [86] and BCN [95]. For more complex sequence dependencies, the external inference module performs deep temporal inference through models such as RNN [7] and Transformer [96]. The detection results are then mapped into rules or policies and distributed back to the programmable data plane, forming a closed-loop process of temporal feature extraction, preliminary detection, collaborative inference, and policy update.

Based on this collaborative mechanism, lightweight temporal detection in programmable switch environments is often implemented through time-window-based state maintenance and online feature updating. Luo et al. proposed P4-IDet, which maintains time-window-based statistical features inside switches and models traffic variation trends online [98]. He et al. used the temporal continuity of synchrophasor data to support event detection on PSs, showing the effectiveness of temporal features in highly real-time industrial networks [86]. These methods can capture key temporal changes without introducing complex models. However, their expressive capability remains limited, making it difficult to characterize long-term dependencies in complex attacks.

Adaptive monitoring further improves the flexibility of temporal feature modeling. Zang et al. proposed a meta-learning-based in-band monitoring method to select and adjust temporal traffic features across different time scales [99]. AMSO-INT uses reinforcement learning (RL) to adapt telemetry strategies according to monitoring requirements and network conditions [100]. These methods reduce dependence on fixed monitoring configurations, but they usually require learning or optimization logic beyond the pure forwarding pipeline.

For complex traffic environments, temporal behavior modeling often requires collaboration between the data plane and external inference modules. Yan et al. proposed Brain-on-Switch, which introduces neural network inference capability near PSs to analyze temporal traffic evolution [96]. Zhao et al. adopted recurrent neural networks in RIDS to model traffic sequences with programmable-switch co-design [7]. Xie et al. proposed Soterv2 to model attack-behavior evolution through cooperation between the data plane and switch CPU [95]. These methods provide stronger temporal representation capability than statistics-based methods, but they also increase system complexity and deployment cost.

Compared with lightweight feature-based detection, temporal behavior modeling improves the ability to capture traffic sequence dependencies and identify persistent or slowly evolving anomalies [7,98]. Behavioral fingerprinting is useful for long-term behavior analysis and root cause identification, but it is difficult to fully implement in the data plane. Time-window-based methods are more deployable, but their effectiveness is sensitive to window length, update frequency, and data dynamics [86,97,98]. Adaptive monitoring can adjust temporal observations according to traffic changes, but it depends on additional learning or optimization mechanisms [99,100]. Neural-network-based temporal models provide stronger sequence modeling capability, but their inference process usually requires switch CPU, external inference modules, or control-plane assistance [7,95,96]. Therefore, temporal behavior modeling is more suitable as a collaborative detection capability for continuous monitoring and attack evolution analysis, rather than as an independent line-rate detection function in the data plane.

3.3 Capability Boundaries and Cross-Technology Integration

Programmable switches provide a practical platform for moving part of the detection process into the data plane, especially for packet parsing, flow measurement, and lightweight anomaly identification. However, their forwarding pipelines are not designed for general-purpose AI execution or complex security reasoning. Therefore, it is necessary to clarify what can be executed inside programmable switches and what should be supported by upper-layer mechanisms. This subsection discusses the capability boundaries of programmable-switch-based detection and further summarizes two representative cross-technology integration patterns: data-plane-supported knowledge-plane architecture and data-plane-supported blockchain architecture.

3.3.1 Capability Boundaries of Programmable-Switch-Based Detection

Although PSs are suitable for low-latency detection, their AI detection capability is constrained by fixed pipeline stages, limited register memory, restricted arithmetic operations, and strict per-packet processing latency [38,101]. These constraints make PSs more suitable for packet parsing metadata extraction [79], counter updates and Sketch-based measurement [82,83], simple state maintenance, and lightweight conditional inference [84,85]. In contrast, complex matrix operations, large neural networks, iterative optimization, and frequent model updates are difficult to execute inside the forwarding pipeline [88].

Therefore, PSs should not be regarded as general-purpose AI execution platforms. Their main role is to provide low-latency sensing [79], compact measurement [82,83], preliminary inference [84,85], and temporal evidence collection [86,97,98]. More complex tasks, such as semantic interpretation, trusted auditing, attack-chain reasoning, model lifecycle management, and explanation generation, should be supported by upper-layer modules, including the control plane, knowledge plane, BC-based trust mechanisms, KG-based reasoning modules, and LLM-based analysis modules. Table 4 summarizes these capability boundaries and cross-technology interfaces.

Table 4: Capability boundaries and cross-technology interfaces of programmable-switch-based detection.

Function Role	Suitable Data-Plane Tasks	Main Limitations	Cross-Technology Interfaces
Feature extraction	Packet parsing, flow counters, header statistics, and compact metadata extraction [79–81]	Limited feature dimensions and weak semantic representation	Provides structured features for KG modeling and LLM-assisted alert explanation [14,37].
Approximate measurement	Sketch-based flow estimation, heavy-flow monitoring, and in-network aggregation [82,83]	Hash collisions and approximate results	Provides compact evidence for BC-based recording and cross-node verification [31,35].
Lightweight inference	Threshold rules, decision trees, random forests, and distilled lightweight models [84,85,89,90]	Limited model complexity and weak generalization to unknown attacks	Sends alerts, risk scores, and preliminary decisions to upper-layer reasoning modules.
Temporal state maintenance	Sliding-window statistics, temporal state updates, behavioral fingerprints, and temporal summaries [86,97,98]	State memory pressure and limited long-term dependency modeling	Provides temporal traces for control-plane analysis and knowledge-plane reasoning [14,95,96].
Collaborative detection	Data-plane sensing, preliminary filtering, selective reporting, and control-plane-assisted inference [88,99,100]	Reporting overhead, control latency, and dependence on upper-layer resources	Supports BC audit, KG-based correlation, LLM-based explanation, and model updating [14,31,35].

Note: DP: Data Plane; KG: Knowledge Graph; LLM: Large Language Model.

3.3.2 Data-Plane-Supported Knowledge-Plane Architecture

Based on these capability boundaries, programmable-switch-based detection can be organized into a data-plane-supported knowledge-plane architecture, as shown in Fig. 7. The architecture consists of three layers: the programmable data plane, the control plane, and the knowledge plane. This organization follows the general idea of AI-driven packet forwarding, where the data plane provides network observations, the control plane translates and distributes policies, and the knowledge plane supports higher-level analysis and decision-making [102].

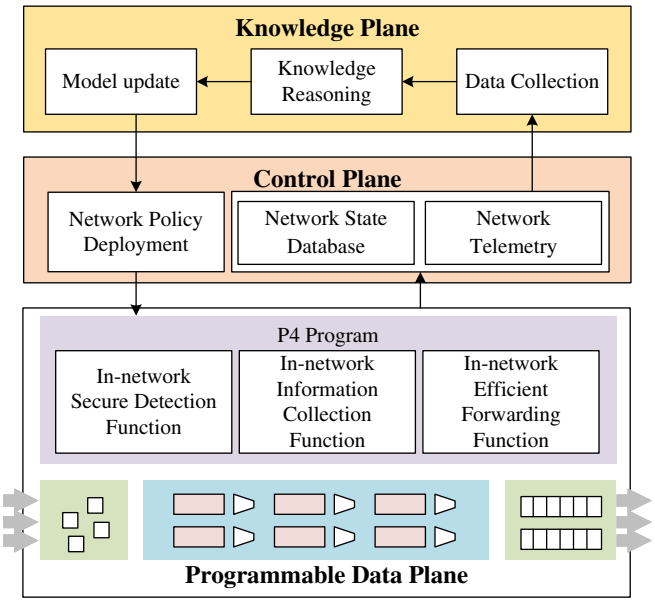

Figure 7: Data-plane-supported knowledge-plane architecture.

At the bottom layer, the programmable data plane executes P4 programs and supports three types of in-network functions. First, in-network secure detection performs lightweight anomaly or attack detection through compact models, compressed data structures, or packet-level analysis [17,103,104]. Second, in-network information collection extracts packet-level, flow-level, and time-series states and aggregates them through Sketches, counters, accumulators, or device fingerprints [103,105,106]. Third, in-network forwarding supports rule matching, queue adjustment, and traffic engineering actions under programmable match-action pipelines [101]. These functions provide timely observations and preliminary processing results for upper-layer analysis.

The control plane connects data-plane observations with knowledge-plane decisions. It collects telemetry results, maintains network states, and deploys policies or model rules generated by the knowledge plane. For example, monitoring sketches can report extracted features to the control plane for drift verification and model adaptation [17], while switch-assisted anomaly detection frameworks can use the control plane or switch CPU to confirm suspicious packets and adjust detection thresholds [104]. In this sense, the control plane mainly serves as an orchestration layer rather than a line-rate detection component.

At the top layer, the knowledge plane organizes telemetry results, compact traffic summaries, device fingerprints, and detection outputs into reusable network knowledge. It can support anomaly interpretation, traffic engineering, policy generation, and model update when traffic distributions or attack patterns change [17,101,104,106]. Through this layered process, data-plane observations can be

converted into knowledge-plane decisions, while updated rules or thresholds can be deployed back to programmable switches.

3.3.3 Data-Plane-Supported Blockchain Architecture

In addition to knowledge-plane reasoning, programmable-switch-based detection can also be integrated with BC or distributed ledger technologies to improve the traceability and cross-domain sharing of detection results, as shown in Fig. 8. This architecture can be organized into three layers: the programmable switch, the control plane, and the BC layer. The programmable switch performs traffic observation and preliminary detection; the control plane aggregates detection metadata and coordinates security policies; and the BC layer records verified evidence and supports cross-domain sharing [31,35].

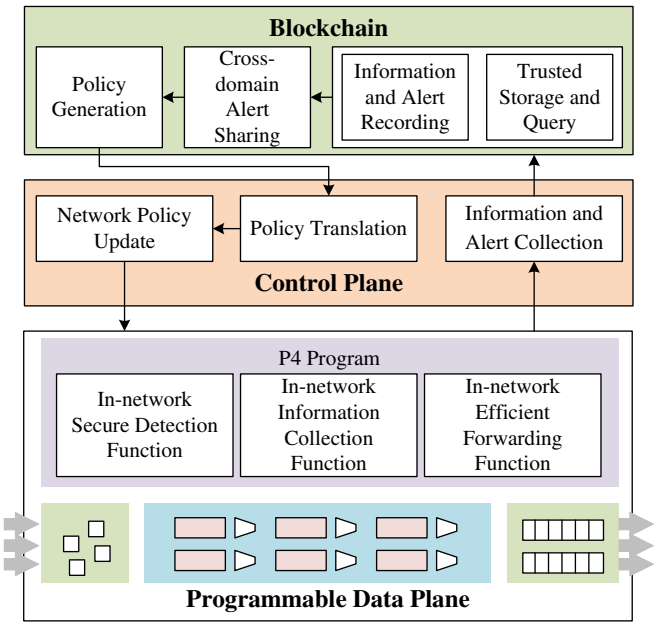

Figure 8: Data-plane-supported Blockchain architecture.

At the bottom layer, the programmable switch parses packets, extracts traffic features, maintains compact flow states, and performs lightweight in-switch detection. The programmable switch can first extract packet-level or flow-level metadata for detection [79], while the detection logic can be implemented through threshold rules, Sketch-based statistics [82,83], decision trees, or other lightweight models compatible with match-action pipelines [84,85]. When suspicious packets or abnormal flow states are observed, the switch generates compact detection metadata, such as alert identifiers, suspicious flow keys, risk scores, packet features, or malicious traffic probabilities, and reports them to the control plane through digest messages, packet-in events, telemetry reports, or switch-control interfaces [107].

The control plane serves as the coordination layer between fast data-plane detection and BC-based recording. It receives alerts and detection metadata from programmable switches, performs aggregation or verification, and generates corresponding detection decisions and mitigation policies. In BC-assisted detection, the control plane can also transform verified alerts, attack indicators, and mitigation decisions into standardized records before writing them to the BC layer. This design separates line-rate packet processing from consensus, cryptographic verification, and trust management, avoiding the deployment of heavyweight BC functions inside the forwarding pipeline.

At the top layer, the BC layer provides tamper-resistant recording, alert sharing, and synchronization of attack indicators or mitigation strategies. In this architecture, data-plane detection results are not directly treated as globally trusted conclusions. Instead, they are first aggregated and verified by the control plane and then recorded or shared through BC-based mechanisms. DPZTN follows this design logic by using a programmable-switch-based ZTNE to monitor authenticated traffic, extract packet features, compute malicious traffic probability, and report the result through the control plane to the blockchain for behavior evaluation and policy adjustment [107]. Therefore, the BC layer mainly enhances the auditability and shareability of detection outputs, rather than participating in line-rate packet processing.

Overall, PSs provide fast evidence acquisition, compact state monitoring, and preliminary anomaly detection, while upper-layer modules support semantic correlation, trusted recording, and management-plane reasoning. This layered design avoids overloading the forwarding pipeline and provides a practical way to combine programmable data planes with knowledge-driven and trust-oriented security mechanisms.

4 Knowledge-Driven AI Detection Methods

Knowledge-driven AI detection aims to bridge the gap between low-level detection outputs and high-level attack semantics. Purely data-driven models can identify statistical differences between normal and abnormal samples, but their outputs are often difficult to associate with attack behaviors, attack stages, vulnerable assets, or response strategies. By introducing explicit security entities, relations, rules, techniques, procedures, and evidence paths, knowledge-driven methods transform isolated alerts, traffic features, and logs into semantically organized knowledge structures. Therefore, their value lies not only in improving detection performance, but also in supporting interpretable detection, alert correlation, attack attribution, evidence organization, and cross-domain knowledge reuse.

This section reviews knowledge-driven AI detection from three connected perspectives. Knowledge representation and construction transform fragmented security data into reusable knowledge bases (KBs) or KGs. Knowledge-enhanced detection and attack reasoning use implicit or explicit knowledge to support malicious traffic detection, behavior mapping, attack-chain reconstruction, and attribution. Capability boundaries and cross-technology interfaces further clarify how knowledge-driven methods cooperate with PSs, BC, and LLMs in practical systems.

4.1 Knowledge Representation and Construction

Security knowledge representation and construction provide the foundation for knowledge-driven detection. Their objective is to transform fragmented security information into structured and reusable knowledge [108]. In AI detection systems, this process usually involves behavior-level abstraction, schema-level organization, dynamic updating, and automated extraction. Behavior-level abstraction maps low-level traffic records, host logs, alerts, and threat intelligence into attack behaviors and stages. Schema-level organization further defines security entities, attributes, relations, and semantic categories. In this context, entities usually refer to objects such as IP addresses, ports, hosts, services, attack behaviors, attack stages, and tactics or techniques, while attributes describe their observable properties, such as packet statistics, protocol fields, temporal features, vulnerability descriptions, and confidence scores. Dynamic updating keeps KBs and KGs effective when new attacks, vulnerabilities, and tactics appear. Automated extraction further reduces the dependence on manual annotation and fixed rules.

As shown in Fig. 9, security KG construction starts from heterogeneous security data, including traffic records, host logs, alerts and detection results, threat intelligence and vulnerability descriptions, and asset and network context. Automated extraction first identifies security entities and attributes, resolves coreference, extracts relations, and links extracted objects to existing knowledge. The acquired knowledge forms a raw

security KG. It is then refined through KG completion, knowledge fusion, and incremental KG completion by using existing KBs, KGs, MITRE ATT&CK, threat intelligence, and historical attack knowledge. Finally, incremental knowledge updating and knowledge dynamics modeling support the evolution of security KGs, enabling the representation of new attacks, new vulnerabilities, and changing behavior patterns.

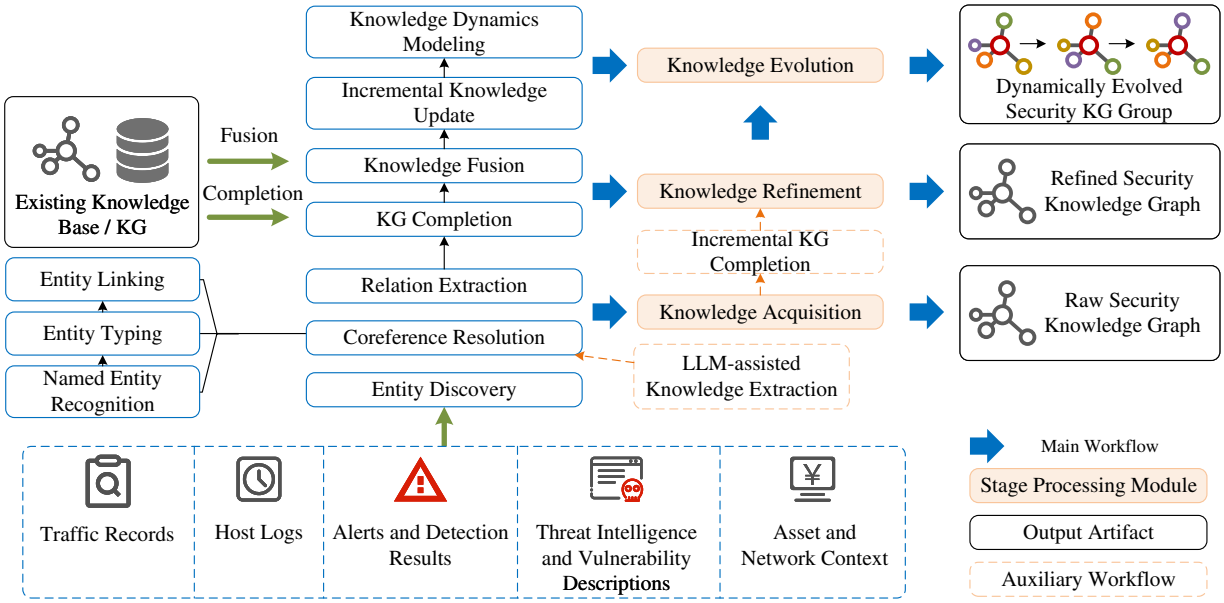

Figure 9: Security knowledge representation and construction process.

At the behavior-abstraction level, researchers introduce concepts such as behavior sequences and attack stages to improve the semantic completeness of KGs. Putra et al. proposed the B-CAT model to analyze attack behavior patterns in network traffic and characterize botnet activities. Its core idea is to aggregate scattered traffic features into behavior-level attack units, which provides a basis for behavior-oriented knowledge representation [109]. In advanced persistent threat scenarios, Byrapuneni and Saidireddy modeled attack stages through multi-label classification. This approach captures the coexistence of attack behaviors across time and strategy dimensions, and supports the modeling of relationships among attacks, stages, entities, and attributes in KGs [110].

At the schema-organization level, the MITRE ATT&CK framework is widely used to normalize attack knowledge. Rajesh et al. analyzed attack detection and simulation processes based on this framework, showing that tactic and technique labels can describe attack chains in a unified manner [111]. Qiu et al. further integrated attack tracing information with TTP knowledge in the Zoomer system and used graph learning to represent advanced persistent threat behaviors. Their findings indicate that explicitly encoding attack stages and behavior relations improves consistency across samples and stages [112]. These studies show that MITRE ATT&CK can serve as a high-level semantic backbone for security KGs by constraining entity types, attribute descriptions, and relation categories.

Dynamic KB and KG updating is necessary for coping with evolving attack environments. Guo et al. proposed a malicious behavior KB centered on attack behavior patterns, supporting dynamic deployment of security services and demonstrating the practical value of KGs in defense orchestration [113]. Feng et al. systematically analyzed the construction process of cybersecurity behavior KBs and summarized key steps such as behavior abstraction, knowledge organization, and storage formats, providing methodological support for

engineering implementation [114]. Wang et al. further emphasized that continuous knowledge updating can compensate for the limitations of purely model-based approaches in unknown attack scenarios [115].

Recent studies further show that knowledge construction is becoming increasingly automated. LLM-assisted methods can extract entities, attributes, relations, and attack behaviors from heterogeneous security data and transform them into security KGs that support querying and reasoning [116–119]. These methods reduce manual construction costs and expand the range of available knowledge sources. Meanwhile, KGs are gradually evolving from static background knowledge into supporting structures for detection, reasoning, traceability, and response decision-making [120].

Table 5 summarizes representative knowledge construction mechanisms and their roles in AI detection. The table focuses on construction roles, detection value, and practical constraints, while the preceding discussion explains how these mechanisms support behavior abstraction, semantic normalization, entity-attribute organization, knowledge updating, and automated extraction.

Table 5: Comparison of knowledge representation and construction mechanisms.

Mechanism	Knowledge Object	Detection Function	Main Advantage	Main Limitation
Attack behavior modeling	Behavior sequences, attack stages, botnet activities, APT stages, and related traffic attributes	Converts traffic or log features into behavior-level attack units	Improves semantic completeness of attack representation	Depends on behavior abstraction granularity and labeled behavior evidence [109]
Unified knowledge structure	Tactics, techniques, procedures, entities, attributes, and relations	Provides a common semantic schema for alerts, attacks, evidence, and contextual attributes	Improves consistency and comparability across systems	Needs scenario-specific extension for satellite, edge, and IoT environments [111]
Dynamic KB/KG updating	New attacks, vulnerabilities, behavior patterns, security services, and evolving entity attributes	Supports knowledge reuse, defense orchestration, and adaptation to new attacks	Reduces dependence on fixed rules and static models	Faces update latency, noisy extraction, and maintenance cost [115]
LLM-assisted construction	Entities, attributes, relations, attack descriptions, and multimodal threat knowledge	Automates knowledge extraction and attack reasoning preparation	Reduces manual construction burden and expands knowledge sources	Requires verification to avoid hallucinated entities, incomplete attributes, or incorrect relations [116–119]

4.2 Knowledge-Enhanced Detection

Malicious traffic detection is evolving from single statistical-feature anomaly identification toward the analysis of complex attack behaviors, unknown threats, and multi-stage attack processes. Data-driven methods still face limitations in generalization, interpretability, and continuous adaptability. Although traditional data-driven methods can learn discriminative features from traffic samples, their outputs often lack explicit semantic support, making it difficult to explain detection results, transfer knowledge across scenarios, and adapt to evolving threats. Therefore, recent studies have introduced security knowledge into detection processes to strengthen the understanding of attack behaviors, attack semantics, and threat evolution.

As shown in Fig. 10, knowledge-enhanced detection can be organized into four closely related routes: implicit knowledge learning, explicit knowledge-assisted detection, behavior semantic mapping, and reasoning-oriented knowledge evolution. The solid arrows indicate the main progression from latent pattern extraction to explicit knowledge support, behavior-level interpretation, and knowledge evolution. The dashed arrows indicate feedback for knowledge refinement and update. Specifically, implicit knowledge learning embeds historical attack patterns into model parameters and representation spaces. Explicit knowledge-assisted detection introduces external KBs or KGs so that attack types, vulnerabilities, assets, behavioral relations, and semantic labels can support feature modeling and detection decisions. Behavior semantic mapping connects traffic records, logs, and user behaviors with attack stages, role relationships, or behavioral chains. Reasoning-oriented knowledge evolution further updates knowledge and detection logic to support unknown attack discovery, concept-drift adaptation, and continuous learning.

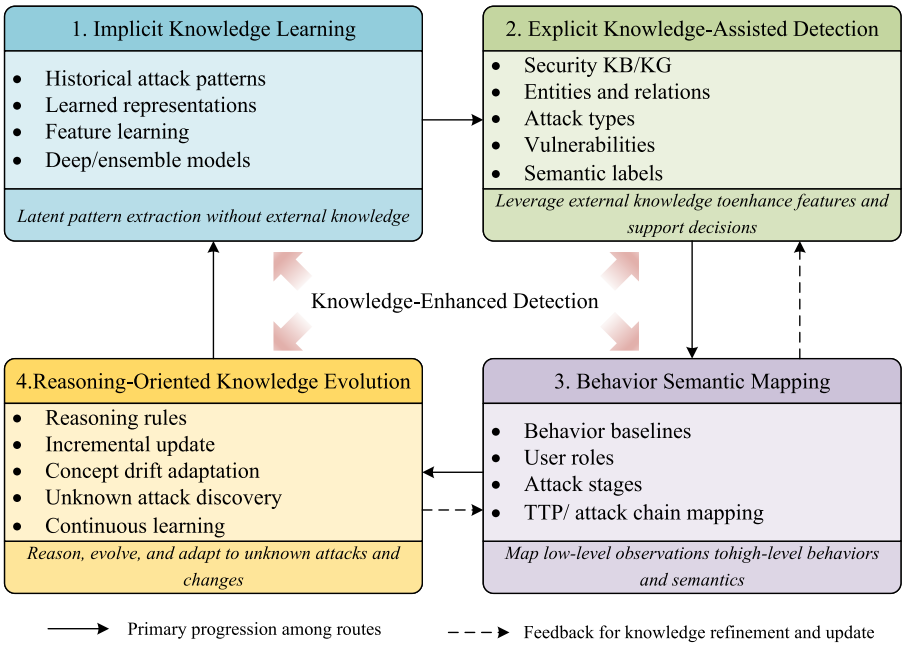

Figure 10: Knowledge-enhanced detection.

Implicit knowledge learning methods do not rely on external KBs or KGs. Instead, they encode historical attack patterns into model parameters and representation spaces through DL, ensemble learning, feature learning, or training mechanisms. These methods extract latent regularities from labeled or unlabeled traffic for malicious traffic classification or anomaly detection. They are relatively easy to deploy and compatible with real-time engineering systems. However, because knowledge is represented implicitly, such models

usually cannot clearly explain their decisions, and the learned knowledge is difficult to reuse, verify, or transfer across scenarios [121].

Methods based on security KBs and KGs extend malicious traffic detection from a purely data-driven process to a collaborative process involving data features and explicit security knowledge. The core feature of such methods is that knowledge exists independently in the form of entities, relations, semantic labels, or graph structures, and directly participates in feature modeling or detection decisions. Kumari and Kumar proposed the KGs-DeepCNN method, which uses a KG to describe the semantic relationships among different attack types, extracts high-order structural features through a graph convolutional network, and then inputs them into a deep classifier for detection [20]. This study shows that explicit knowledge structures can compensate for the limited ability of raw traffic features to express semantic correlations. Zhang et al. also adopted graph structures to model user behavior relationships in their study on user representation learning and fraud detection, providing a reference for the application of KGs in abnormal behavior identification [122]. In addition, some systematic studies have pointed out that security KBs can be used to uniformly store attack patterns, vulnerability information, and semantic labels, thereby supporting cross-scenario detection and rule reuse. BRIDG-ICS targets industrial control system scenarios by integrating assets, vulnerabilities, attack behaviors, and risk indicators into an industrial security KG, thereby supporting threat analysis and resilience assessment for specific industrial scenarios [123]. Compared with implicit learning, explicit knowledge-assisted detection is more favorable for rule reuse, cross-scenario correlation, and result interpretation, but its effectiveness depends on knowledge construction quality, domain coverage, and update capability.

Behavior semantic mapping methods focus on the correspondence between low-level observations and high-level attack behaviors. Malicious traffic is usually not an isolated statistical anomaly, but an external manifestation of adversarial behaviors in the network. Therefore, traffic volume, packet interval, connection frequency, and flow duration are often insufficient to explain attack intent. Legg et al. constructed behavioral baselines based on user and role profiles and identified anomalies through deviation analysis, effectively mapping low-level logs to high-level behavioral semantics [124]. Kim et al. used user behavior modeling and anomaly detection to identify insider threats, emphasizing behavior patterns over single traffic instances [125]. Sharma et al. applied long short-term memory (LSTM) autoencoders to model user behavior sequences and used reconstruction error to measure anomalies. This reflects the idea of mapping temporal behavior into latent semantic states [126]. Ye and Han used hidden Markov models to describe transitions between user behavior states, improving semantic interpretability [127]. Yamauchi et al. demonstrated that behavior semantic modeling has good cross-scenario applicability [128]. Böse et al. proposed the RADISH system, which identifies anomalies through real-time multi-source behavior analysis, highlighting the value of process-oriented attack modeling [129]. These methods elevate detection results from traffic-level anomalies to behavior-level or attack-stage-level anomalies, improving both interpretability and security analysis value.

Reasoning-oriented knowledge evolution methods emphasize continuous adaptation to emerging threats. In real network environments, attack strategies, traffic distributions, and system states change over time, so static KBs and fixed detection models are difficult to maintain long-term effectiveness. Zhao et al. proposed the Trident framework to address unknown traffic and concept drift. It uses incremental learning and unknown attack detection to gradually integrate new attack types into the system [130]. Chen et al. proposed an improved density peak clustering algorithm for unsupervised detection, allowing models to adapt to changing traffic distributions [131]. Li et al. and Huang et al. studied distributed learning and knowledge evolution across heterogeneous nodes. Their work addresses data heterogeneity and supports cross-domain detection [132,133]. In federated learning scenarios, Meng et al. further explored abnormal

client detection and knowledge inference for malicious participants [134]. The core of this route is not only to use existing knowledge for detection, but also to update the knowledge system through detection feedback, enabling continuous learning, dynamic reasoning, and cross-domain collaborative analysis.

Table 6 summarizes the main routes of knowledge-enhanced detection. Overall, knowledge-enhanced detection does not replace traditional AI-based detection, but complements data-driven methods at different levels. Implicit knowledge learning supports rapid detection and engineering deployment. Explicit knowledge-assisted detection enhances semantic representation and rule reuse. Behavior semantic mapping improves result interpretability. Reasoning-oriented knowledge evolution supports adaptation to unknown attacks and dynamic environments. However, future studies still need to address the high cost of knowledge construction, limited cross-scenario transferability, delayed knowledge updating, and trustworthy verification of reasoning results.

Table 6: Comparison of knowledge-enhanced detection methods.

Route	Knowledge Carrier	Detection Role	Advantages	Limitations
Implicit knowledge learning	Model parameters, learned representations, or historical attack patterns	Learns hidden attack patterns from labeled or unlabeled traffic	Easy to deploy; suitable for real-time classification and evolving traffic	Weak interpretability; knowledge cannot be explicitly reused
Explicit knowledge-assisted detection	Entities, relations, semantic labels, attack-type associations, and asset relations	Enhances feature representation and detection decisions through explicit knowledge	Better interpretability and cross-scenario reuse	High construction cost; limited coverage of unknown attacks [20,123]
Behavior semantic mapping	User roles, behavior baselines, attack stages, and TTP mappings	Maps low-level observations to high-level behavioral semantics	Useful for insider threats, APTs, and staged attacks	Sensitive to baseline drift and incomplete attack frameworks [124]
Reasoning-oriented knowledge evolution	Incremental knowledge, distributed learning states, confidence scores, and reasoning rules	Updates knowledge and detection logic under new attacks or concept drift	Supports adaptation and continuous learning	Symbolic reasoning is still limited; online reasoning cost remains high [132]

4.3 Knowledge-Driven Attack Reasoning and Attribution

Knowledge-driven detection is closely related to attack reasoning and attribution. Detection identifies abnormal traffic or behavior, while reasoning and attribution explain the attack source, propagation path, attack stage, correlated alerts, and attribution explanation. Early traceback studies mainly relied on network-layer forwarding path information. They embedded path identifiers in packets or network devices to trace

the attack source [135]. As attacks evolve from traffic flooding to multi-stage and cross-system operations, attribution increasingly needs to fuse forwarding evidence, attack semantics, entity relations, temporal evidence, and learning-based representations. KG-based entity–relation modeling can provide structured support for organizing security entities and analysis processes [136].

As shown in Fig. 11, knowledge-driven attack reasoning and attribution can be viewed as an evidence-fusion process. Path-based traceback provides packet marking, path encoding, logging, path reconstruction, and forwarding evidence. Semantic and rule-based reasoning organizes alerts through attack stages, MITRE ATT&CK mappings, correlation rules, and attack-process reconstruction. KG- and relation-based attribution models security entities, attack relations, provenance graphs, threat intelligence, and relational evidence. Learning–reasoning integration introduces graph learning, temporal modeling, LLM-assisted reasoning, evidence evaluation, and explainable attribution. These heterogeneous inputs are aligned in an evidence fusion and reasoning core, where attack-chain reconstruction, confidence update, and attribution logic jointly support attribution outputs, including attack source, propagation path, attack stage, correlated alerts, and attribution explanation.

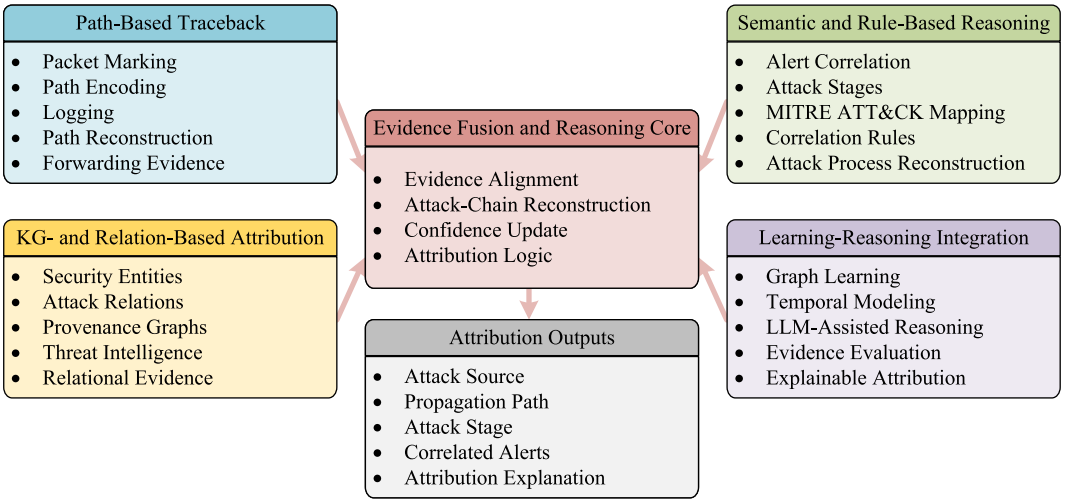

Figure 11: Knowledge-driven attack reasoning and attribution.

Existing attribution studies can be organized according to evidence type and reasoning mechanism. Path-based methods focus on forwarding evidence. Semantic and rule-based methods organize alerts according to attack stages and correlation rules. KG- and relation-based methods represent attack-related objects as entities and relations. Learning–reasoning integration methods use graph learning, temporal modeling, or LLM-assisted reasoning to improve attribution under complex evidence conditions.

Attack traceback methods based on path and network structure knowledge mainly aim to recover the forwarding path of attack traffic. Common mechanisms include probabilistic marking, path encoding, logging, and path reconstruction. Park and Lee analyzed probabilistic packet marking in denial-of-service scenarios and showed that marking probability and attack traffic volume significantly affect tracing success. In large-scale networks, this approach has been extended to the autonomous system level, trading path accuracy for scalability [137]. To reduce dependence on large volumes of attack packets, some studies propose single-packet traceback mechanisms [138]. Hybrid schemes have also been developed to support heterogeneous environments such as IPv4 and IPv6 coexistence [139]. Jeong and Lee introduced compressed hash tables and aggregation nodes, combining logging and data mining techniques for path reconstruction [140].

In software-defined networking environments, path-observation-based probing mechanisms provide more accurate forwarding evidence for traceback [141].

Traceback methods based on attack semantics and rule knowledge regard attacks as behavioral processes composed of multiple stages and techniques. Castillo-Fernández et al. proposed a multi-layer alert correlation framework that aggregates events and alerts to construct high-level attack descriptions [142]. Muñoz-Calle et al. developed an alert correlation method based on the MITRE ATT&CK framework. It maps low-level alerts to attack techniques and stages, enabling structured representation and traceback analysis [143]. Chen et al. used multi-label modeling to represent relationships among different attack stages, supporting semantic reconstruction of attack paths [144]. Bijalwan et al. constructed evidence graphs and applied reasoning mechanisms to infer attacker intent and perform traceback analysis [145]. These methods strengthen attribution by linking isolated alerts with attack stages, TTPs, correlation rules, and attack-process reconstruction.

KG- and relation-based security analysis methods model attack-related objects, security tools, assets, and evidence as entities and relationships. Sun et al. constructed a security tool KG, showing how entity–relation modeling can represent security capabilities and analysis processes [136]. In threat attribution, hierarchical relation-driven IP attribution methods aggregate evidence across different relational levels, enabling systematic tracing of attack sources [146]. Xiao et al. applied relational modeling in industrial control systems by abstracting attack data into structured relations. Their approach supports malicious IP traceback [147]. Recent studies further combine provenance graphs and external threat intelligence. Therefore, KG- and relation-based methods are suitable for aligning security entities, attack relations, provenance evidence, and threat intelligence across heterogeneous sources.

Attack traceback methods based on learning–reasoning integration enhance evidence association by introducing ML models and improve traceback reliability by combining knowledge constraints or reasoning mechanisms. Hadem et al. combined intrusion detection with selective logging, providing high-value evidence for subsequent traceback [148]. Early studies also explored multi-agent systems and self-organizing maps for attacker tracing. In network forensics and trusted analysis, trustworthy AI methods have been applied to support attack detection and evidence evaluation [149]. Temporal attention-based explainable models improve the understanding of attack evolution [150]. This route combines graph learning, temporal modeling, LLM-assisted reasoning, evidence evaluation, and explainable attribution to improve reasoning under complex multi-source evidence.

Table 7 summarizes representative knowledge-driven attribution routes. The table summarizes the evidence basis, attribution use, strengths, and limitations of each route, while Fig. 11 further illustrates how heterogeneous evidence is fused for attack-chain reconstruction, confidence update, attribution logic, and explanation-oriented outputs.

Table 7: Comparison of knowledge-driven attack attribution routes.

Attribution Route	Evidence Basis	Attribution Use	Strengths	Limitations
Path and structural knowledge	Packet marks, forwarding paths, logs, and network topology	Recovers attack forwarding paths in traffic-dominated attacks	Clear path meaning; intuitive traceback result	Sensitive to routing dynamics, multi-source attacks, and forged paths [138]

(Continued)

Table 7 (continued)

Attribution Route	Evidence Basis	Attribution Use	Strengths	Limitations
Semantic and rule knowledge	Alerts, attack stages, TTPs, and correlation rules	Maps alerts to attack processes and reconstructs multi-step attacks	Strong interpretability; maps alerts to attack processes	Depends on complete rules and attack models [142]
KG and relational modeling	Attack entities, tools, assets, vulnerabilities, and relations [136]	Aligns cross-source evidence and supports threat attribution	Supports relational reasoning and cross-domain knowledge reuse	High construction cost; relation noise and knowledge timeliness problems [146]
Learning–reasoning integration	Graph embeddings, temporal evidence, rules, structured queries, and provenance evidence	Improves correlation capability under complex multi-source evidence	Better correlation capability and flexible reasoning	Generalization and auditability remain challenging [149]

4.4 Capability Boundaries and Cross-Technology

Knowledge-driven detection improves semantic understanding, interpretability, and correlation analysis, but it is not a replacement for real-time traffic detectors. Its capability boundary is mainly determined by knowledge construction cost, reasoning latency, knowledge freshness, and the trustworthiness of shared knowledge. In high-throughput or latency-sensitive scenarios, complex KG reasoning cannot be performed for every packet or flow. Therefore, knowledge-driven methods are more suitable as a semantic analysis and reasoning layer above lightweight detection modules.

As network systems evolve from single-domain and centralized architectures to cross-domain, distributed, and autonomous forms, the organization, sharing, and utilization of security knowledge are also changing. Traditional centralized KBs are gradually being complemented by cross-domain distributed knowledge architectures. These architectures emphasize distributed storage, collaborative updating, and trusted use of security knowledge across different network domains, system layers, and functional entities [11]. Existing studies show that distributed control, semantic modeling, federated learning, and BC mechanisms can support cross-domain knowledge collaboration and improve the robustness and scalability of security systems [9,10,15]. However, distributed knowledge sharing still faces trust risks, model poisoning, update latency, and coordination overhead [26,151,152].

The role of knowledge-driven detection also differs across communication scenarios. Cloud networks require scalable knowledge management and cross-tenant alert correlation. Edge networks require lightweight local knowledge and efficient cloud–edge synchronization. Satellite networks require delay-tolerant and hierarchy-aware knowledge sharing under dynamic topology and intermittent links. IoT environments require device-aware knowledge abstraction and distributed reasoning among terminals, gateways, and edge nodes. Therefore, the construction granularity, update frequency, and reasoning location

of KBs and KGs should be adjusted according to scenario-specific latency, resource, and connectivity constraints.

Knowledge-driven detection should also be positioned within a multi-technology collaborative architecture. Programmable switches and edge-side detectors can provide real-time traffic features, alarms, and temporal summaries, but they cannot fully explain attack semantics. KGs can map these outputs to entities, behaviors, attack stages, and evidence relations. BC can record key knowledge versions, evidence indexes, and reasoning results to improve trustworthiness and auditability. LLMs can use KGs, alerts, logs, and trusted evidence as structured context to generate explanations, summarize attack chains, and support decision-making.

Overall, knowledge-driven detection links low-level detection outputs with attack semantics, attack stages, and entity relations. Its main value lies in interpretability, correlation analysis, traceback support, and cross-domain knowledge reuse. However, KB completeness, knowledge freshness, reasoning latency, construction cost, and trustworthiness of shared knowledge remain unresolved issues. A feasible deployment path is to use PSs or edge-side detectors for online screening, KGs for semantic correlation and attribution, BC for trusted evidence management, and LLMs for high-level explanation and decision support.

5 BC-Enabled AI Detection Methods

As distributed networks continue to scale, data sharing, model collaboration, and detection-result feedback introduce increasing trust and privacy risks. Centralized security mechanisms suffer from single points of failure and struggle to meet privacy compliance and accountability requirements in multi-party and cross-domain environments. In this context, blockchain (BC) is introduced as a trusted infrastructure for AI-based detection systems, providing decentralized consensus and tamper-resistant records to support secure coordination and auditability. Detection and inference are still performed off-chain, while BC is used to enhance the trustworthiness of interactions and results.

Therefore, BC-enabled AI detection is formulated as an off-chain detection and on-chain verification architecture. In this framework, AI models perform traffic analysis, anomaly detection, and collaborative inference off-chain, while BC records key evidence including detection outputs, model states, policy updates, access logs, and audit trails.

As illustrated in Fig. 12, this section organizes BC-enabled AI detection according to its role in the detection chain. BC provides trusted support for distributed collaboration by ensuring data provenance, integrity of model updates, and verifiable cross-node coordination. It further enables secure mechanisms such as identity authentication, access control, and reputation management. In addition, BC supports auditability by maintaining immutable records of detection results, evidence chains, model evolution, and policy changes. This design highlights that BC is primarily suitable for low-frequency trust coordination and audit-oriented tasks, rather than real-time traffic detection.

5.1 Distributed Detection and Privacy Protection

In distributed environments, AI detection needs to handle multi-node data sharing, model collaboration, and result feedback. These processes may introduce risks such as data tampering, privacy leakage, and loss of trust in collaboration. BC can enhance the integrity and auditability of detection data and model interactions through decentralized ledgers, tamper-resistant records, and consensus mechanisms. However, BC mainly verifies selected detection evidence and collaboration records, rather than replacing off-chain AI inference.

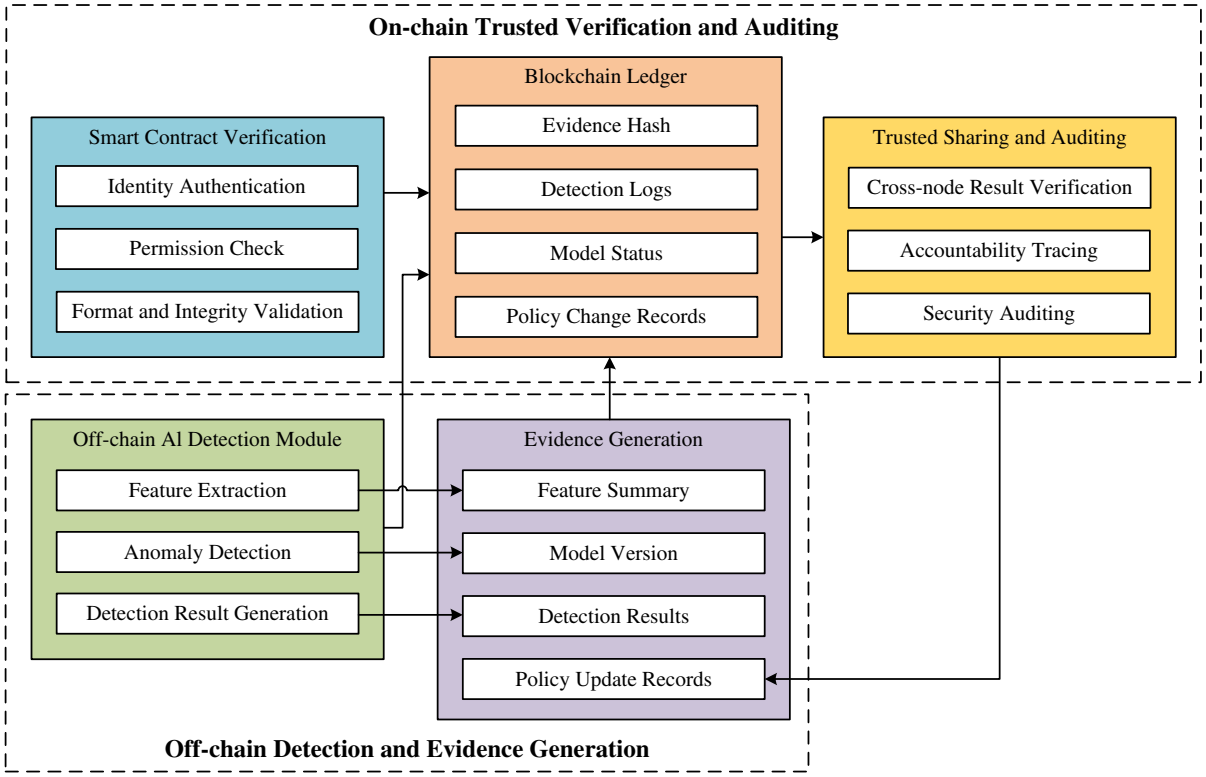

Figure 12: Blockchain-enabled AI detection.

BC-based distributed traffic detection is mainly used to address insufficient trust, data silos, and single points of failure in centralized or weakly collaborative intrusion detection. These methods are usually built on distributed IDSs. They write detection results, feature summaries, model updates, or security knowledge to the BC. This improves the trustworthiness and robustness of cross-node detection collaboration [153].

Existing studies mostly use BC as a trust support layer among distributed detection nodes. One type of work writes alert information, model updates, or security knowledge to the BC as transactions. This allows different nodes to verify the source and integrity of shared information. For example, Sakraoui et al. proposed a BC-enhanced secure knowledge transfer mechanism for 6G scenarios, where distributed detection results are propagated as trusted knowledge [154]. Another type of work combines BC with distributed AI models, using on-chain records to constrain model collaboration, result fusion, and threat intelligence sharing [155–159].

Some studies also focus on tamper resistance and collaborative decision-making. In IoT, Internet of Medical Things, cloud, zero-trust, and Industrial Internet scenarios, detection results or feature summaries can be written to the BC to support cross-node confirmation, response auditing, and traceback [160–164]. From the perspective of method routes, existing studies include rule- or signature-centered approaches, such as BC-based collaborative signature detection and Snort-based distributed detection [165–167], as well as data-driven approaches that deploy ML or DL models on distributed nodes while using BC to verify model updates and detection outputs [168–171].

These studies show a clear boundary of BC-supported distributed detection. Traffic analysis, feature extraction, and model inference are still performed off-chain. BC mainly records low-frequency and high-value information, such as alerts, model versions, feature summaries, rule updates, and evidence indexes. If every flow-level or packet-level result is written on-chain, consensus latency and storage overhead will make the system unsuitable for real-time detection.

In open environments such as the IoT, the Internet of Vehicles, and the Industrial Internet, AI detection data may contain service information, device states, and user behavior trajectories. If these data are directly collected and processed in a centralized manner, the risk of privacy leakage may increase. Therefore, how to reduce data exposure while maintaining detection performance is an important issue in BC-enabled AI detection [172,173].

Existing studies mainly focus on data localization and collaborative learning. Federated learning is widely used to avoid direct sharing of raw data across nodes. BC is often introduced as a trusted coordination and auditing layer in this process. It records model updates, node behavior, and contribution information, helping prevent model poisoning and parameter tampering [174,175]. Some studies further focus on privacy-preserving detection in lightweight IoT or IoMT environments. In these methods, raw device data and behavior records are not directly exposed to all participants. Instead, blockchain is used to record selected evidence, verification results, or misbehavior-related information, thereby supporting collaborative detection while reducing privacy leakage risks [176]. These methods provide stronger privacy protection, but they also introduce additional computational and communication overhead. Therefore, they are mainly suitable for lightweight models, selective collaboration, or offline detection tasks in resource-constrained environments [177].

Other studies address privacy from the perspective of data sharing and access control. Permissioned or consortium BCs are used to manage access rights for detection data, feature summaries, and model outputs. Smart contracts regulate data usage and detection workflows [178,179]. In this framework, BC ensures data provenance and auditability, while AI models operate as controlled off-chain services. These approaches improve compliance in cross-organizational detection scenarios, but their effectiveness still depends on the design of detection models, feature representations, and access policies [171,180].

5.2 Trusted Mechanisms and Auditability

BC-enabled trust mechanisms provide identity authentication, access control, reputation evaluation, and audit support for AI-based detection systems. These mechanisms do not directly improve detection accuracy, but enhance the reliability of detection inputs, collaboration processes, and decision-making outputs.

As shown in Fig. 13, in heterogeneous network environments integrating satellite networks, mobile networks, edge computing, and IoT, entities such as users, devices, gateways, and service nodes require reliable identity authentication. Existing studies have proposed identity-based, certificateless, hybrid cryptographic, and lightweight key management schemes to reduce terminal overhead while ensuring security [181–184]. With the introduction of blockchain technology, identity credentials, authentication results, and verification evidence can be stored on-chain, thereby reducing reliance on centralized authentication authorities and enabling cross-domain verification [185,186]. Furthermore, some studies combine blockchain with edge or fog computing and introduce zero-knowledge proofs to support continuous authentication and fast handover [187]. However, most existing approaches still rely on static authentication mechanisms and are insufficient in capturing dynamic risk variations.

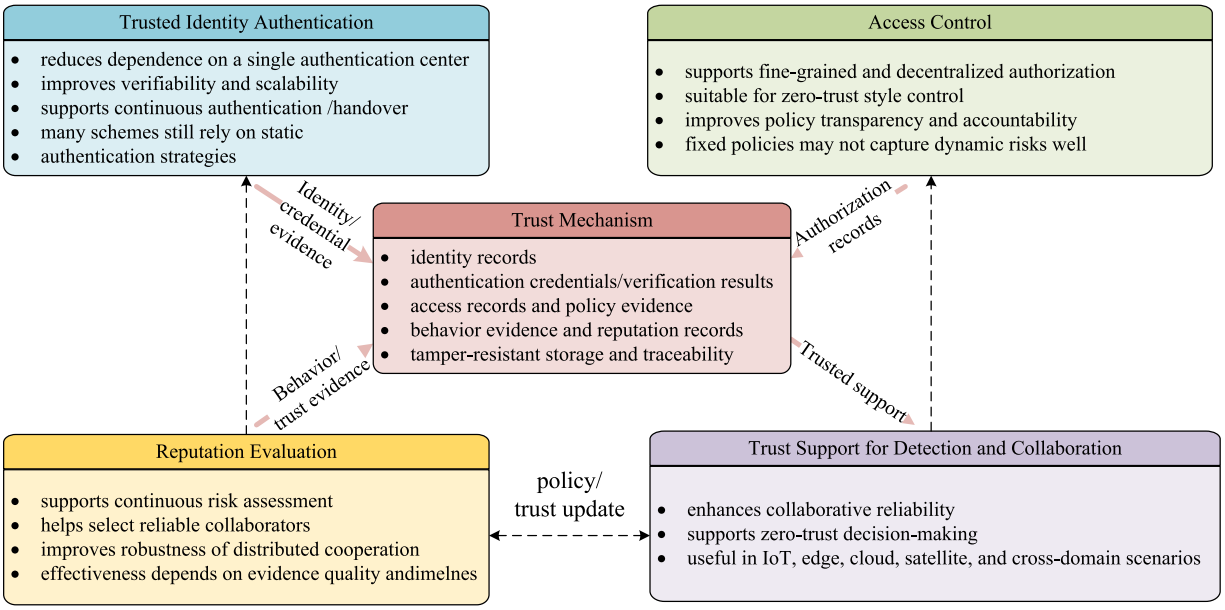

Figure 13: Simplified structure of trust mechanisms in blockchain-enabled AI detection.

Access control mechanisms are designed to enforce authorization policies under zero-trust principles. Existing approaches incorporate contextual information, security attributes, and environmental conditions to support dynamic authorization decisions [188]. Fine-grained control is achieved through continuous authentication and adaptive policy refinement [189]. Blockchain enhances access control by recording authorization policies, decisions, and audit logs in a tamper-resistant manner. Smart contract-based role-based access control (RBAC) and attribute-based access control (ABAC) enable decentralized and verifiable permission management [190–193]. Nevertheless, most existing solutions still depend on static or stage-based attribute evaluation and lack continuous behavior-aware authorization.

Reputation evaluation mechanisms are used to assess node behavior and support malicious node identification and trust-aware decision-making. Centralized reputation systems rely on trusted authorities to collect feedback and compute trust scores [194–197]. In contrast, distributed reputation mechanisms aggregate behavior feedback among nodes and are widely applied in IoT, vehicular networks, and edge computing environments [198–201]. Blockchain further improves reputation evaluation by recording behavioral evidence, interaction logs, and trust assessments on-chain, enabling transparent and verifiable reputation computation [202–206]. However, fixed evaluation intervals and static update strategies still limit the detection of intermittent and adaptive malicious behaviors.

Trust support for detection and collaboration provides secure cooperation among distributed AI detection nodes. Blockchain ensures that identity verification results, authorization evidence, and shared detection outputs are verifiable and tamper-resistant, thereby enabling trustworthy collaboration among nodes. Detection outputs such as alerts, feature summaries, and model states can be recorded on-chain to support accountability and auditability [207]. In multi-stage detection systems, intermediate inference results, feature evolution, and attack progression can be linked into evidence chains, enabling more reliable decision-making in complex attack scenarios [12,13,208]. In addition, combining evidence chains with explainable AI techniques further improves the interpretability and traceability of detection results [209].

At the core of the framework, the blockchain-supported trust mechanism layer provides tamper-resistant storage and verifiable audit support. It records identity information, authentication evidence, authorization policies, behavioral logs, and detection-related audit data, ensuring traceability and integrity of trust evidence across distributed nodes. Overall, the proposed architecture forms a closed-loop trust system in which identity authentication and access control regulate system participation, reputation evaluation provides behavioral trust assessment, and blockchain guarantees verifiable storage and auditability for distributed AI detection systems.

5.3 Capability Boundaries and Cross-Technology

Although BC can provide trusted evidence storage, access constraints, and audit support for distributed AI detection, it still faces clear performance and engineering limitations in practical deployment. The BC consensus process introduces additional confirmation latency. Its throughput is usually difficult to match the requirements of high-frequency network traffic detection scenarios. Therefore, BC should not be positioned as a real-time detection execution layer. It is more suitable for trusted coordination, low-frequency evidence recording, and post-event audit support [210–212].

BC storage and communication overhead increase rapidly as the number of detection nodes, evidence-recording frequency, and cross-domain interactions grow. If all detection results are recorded on-chain, ledger expansion and lower query efficiency may occur. If only summary information is recorded, the completeness of post-event auditing and attack reproduction may be reduced [213]. Therefore, BC-enabled AI detection needs to balance evidence storage granularity, audit requirements, privacy protection, and system overhead.

BC deployment is also constrained by collaboration mechanisms. Cross-domain detection scenarios usually involve different management entities. On-chain data formats, identity authentication mechanisms, access permissions, smart contract security, and responsibility attribution all need unified design. If there is no trusted off-chain data collection mechanism, tamper-resistant on-chain records still cannot guarantee that the original detection data are true and reliable [214,215]. Therefore, BC is more suitable as a trusted coordination and audit support layer in distributed AI detection systems, rather than as a real-time detection execution layer.

The deployment value of BC differs across communication scenarios. Cloud networks require trustworthy model collaboration, cross-tenant audit, and access control. Edge networks require lightweight identity authentication, model update verification, and cloud–edge evidence synchronization. Satellite networks require delay-tolerant trust coordination and space–ground evidence preservation. IoT environments require distributed authentication, privacy-preserving learning, and reputation evaluation for massive heterogeneous devices.

BC also provides interfaces to the other technical routes reviewed in this paper. Programmable switches can generate compact traffic summaries, alerts, and evidence indexes for BC recording. Knowledge graphs can organize detection evidence into semantic units before selected results are written on-chain. LLMs can use on-chain evidence indexes, model versions, and trusted logs to generate verifiable explanations and response suggestions. In the reverse direction, BC can provide trusted evidence constraints for KG reasoning and LLM-based explanation generation.

Overall, BC should be positioned as a trust, coordination, and audit-support layer for distributed AI detection. It is suitable for recording summaries, model states, policies, identity behaviors, reputation updates, and key evidence. It is not suitable for high-frequency real-time detection execution because

of consensus latency, storage cost, and communication overhead. A practical BC-enabled AI detection architecture should follow the principle of off-chain detection, on-chain verification, and cross-layer audit.

6 AI Detection Methods Based on LLMs

With the development of large-model technologies, security detection has begun to integrate large-scale pretraining and general representation learning capabilities. Large models demonstrate strong advantages in semantic understanding, contextual modeling, and reasoning generation, enabling the analysis of multi-source heterogeneous security data. Unlike traditional detection methods that map features directly to labels, LLMs can support attack semantic association, behavior evolution analysis, and explanation of detection results. Therefore, LLMs promote the evolution of security detection from simple classification toward semantic reasoning, attack-chain summarization, and decision support. However, LLMs introduce high inference cost, long context-processing latency, output instability, and potential security risks. Therefore, LLM-based detection should be positioned as an upper-layer semantic reasoning and explanation module rather than a direct replacement for lightweight detection mechanisms.

As shown in Fig. 14, this section reorganizes LLM-enabled AI detection according to its functional role in the detection chain. LLMs operate as upper-layer reasoning components that take as input detection outputs, knowledge-graph representations, and blockchain-verified evidence. They support representation and contextual modeling through traffic sequence modeling, pretraining knowledge encoding, and hierarchical semantic abstraction. In addition, LLMs enable reasoning-based detection through knowledge-guided inference, data completion, attack-chain reconstruction, and explanation generation. Finally, LLMs provide decision-level support by integrating multi-source information and producing human-readable security reasoning results.

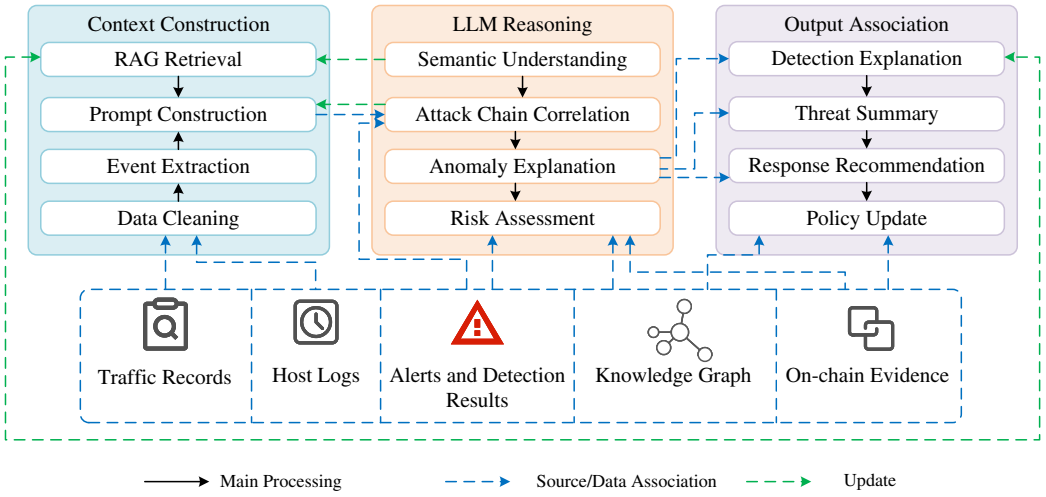

Figure 14: Framework of LLM-enabled AI detection.

6.1 Representation and Contextual Modeling

As encrypted traffic increases and network environments become more dynamic, traditional detection methods struggle to adapt to new scenarios. Large pre-trained models, such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT), have shown strong capabilities in feature extraction, contextual modeling, and transfer learning. This has driven increasing interest in applying large-model ideas to network traffic analysis, log understanding, and intrusion detection.

LLMs can leverage massive unlabeled data through self-supervised learning. This enables them to learn more general traffic representations and improve detection performance and generalization. Their reasoning and generation capabilities can also support data augmentation, adversarial sample generation, and explanation of detection results. Recent studies use the contextual modeling ability of LLMs to perform end-to-end learning on network traffic, logs, or traffic-like sequences. These methods have been applied to DDoS detection, encrypted traffic classification, and malicious traffic identification. For example, BERT-based intrusion detection methods enhance the contextual modeling capability of logs or traffic sequences through pre-trained semantic representations [216].

As network communication becomes increasingly encrypted, the explicit semantic information directly available in traffic decreases. Traditional methods based on statistical features or manual rules have limitations in representing complex attacks and generalizing across scenarios. Therefore, researchers have begun to treat network traffic detection as a sequence modeling and contextual understanding task. Inspired by natural language processing (NLP), packets, flows, or sessions are represented as discrete token sequences and then modeled by Transformer-like architectures. Early studies validated this idea by treating raw payload byte sequences as tokens and inputting them into Transformer models for classification [217]. Although these methods still have limitations in semantic abstraction and cross-scenario generalization, they demonstrate the feasibility of applying large-model architectures to network traffic data.

After confirming the feasibility of traffic language modeling, research has shifted toward robust representation learning under limited labeled data. Encrypted traffic lacks fine-grained semantic information, which makes supervised learning difficult. To address this problem, many studies introduce self-supervised pretraining. ET-BERT pretrains on burst sequences of packets and significantly improves encrypted traffic classification performance [218]. This marks the transition from direct supervised learning to the pretraining–fine-tuning paradigm. At the same time, Transformer-based contextual modeling alone cannot fully capture temporal dynamics in traffic behavior. At the same time, Transformer-based contextual modeling alone may still be insufficient for capturing complex temporal and structural patterns in encrypted traffic. Some methods therefore introduce attention-based deep learning architectures to improve encrypted traffic classification, as shown in ATVITSC [219].

As model capabilities improve, token design and modeling granularity become key factors affecting performance. Simple byte-level or field-level tokens cannot fully capture the hierarchical structure and semantic boundaries of network protocols. To address this limitation, some studies incorporate domain knowledge into tokenization and embedding. FlowletFormer segments traffic into flowlets based on time intervals and introduces protocol-level embeddings, thereby enhancing the Transformer's ability to capture protocol semantics [220]. MIETT treats a flow as a set of packet instances and applies multi-instance learning and hierarchical attention to capture both local packet features and global flow behavior [221]. These studies show that hierarchical and structure-aware modeling is important for complex encrypted traffic understanding.

Recent studies further explore semantics-enhanced detection frameworks. For IoT cyberattack detection, lightweight decoder-only LLMs have been combined with structured traffic-to-text representation, QLoRA-based parameter-efficient fine-tuning, and retrieval-augmented generation mechanisms [222]. Some studies combine lightweight BERT with meta-classifiers for IoT intrusion detection to reduce deployment complexity while maintaining detection performance [223]. DoLLM transforms discrete and unordered traffic events into structured sequences through serialization and tokenization, showing potential in cross-scenario transfer and zero-shot detection [224]. Domain-aware tokenization and large-scale pre-training have also been extended to IoT traffic analysis to improve generalization across new devices and scenarios [225].

To address the limitations of single-flow sequence modeling, researchers introduce semantic enhancement and relational modeling. Traffic data can be aligned with natural-language descriptions, where attack intent and security knowledge serve as semantic priors, as shown in PacketCLIP [226]. Some studies further enhance robustness and interpretability through semantic subspace partitioning or graph-based modeling, such as ASNet and TransGraphNet [227,228].

6.2 Reasoning-Based Detection and Explanation Generation

As the representation learning and generation capabilities of LLMs improve, network traffic detection has begun to expand from discriminative classification to an integrated framework that combines generation, reasoning, and explanation. Such methods are important for data scarcity, class imbalance, concept drift, zero-day attack detection, and analyst-facing decision support. Existing studies can be organized into generative traffic modeling, structure- and semantics-constrained generation, diffusion-based continuous adaptation, and agent-based reasoning and explanation.

Early generative methods aim to model the distribution of real network traffic using LLMs. These models not only perform classification but also generate high-fidelity synthetic traffic data. Such approaches typically use autoregressive or autoencoding frameworks. They tokenize traffic at the flow or packet level and learn representations through generation tasks. TrafficGPT converts flow-level traffic into reversible token sequences, allowing the model to perform classification and generate reconstructable traffic data [229]. Studies show that generative pretraining can improve classification by learning richer traffic distributions, while generated traffic can support data augmentation and simulation.

As generative models improve, researchers find that unconstrained generation can cause semantic drift and structural distortion. This problem is more severe in encrypted traffic and complex protocol scenarios. To address this problem, some studies introduce protocol structures, feature dependencies, and domain knowledge into the generation process. Zhao et al. proposed GBC, which combines protocol-aware tokenization with self-supervised pretraining to better preserve protocol syntax during traffic generation and improve the stability of downstream detection performance [230]. In addition, some studies combine generative models with graph-based modeling and domain knowledge to explicitly model feature dependencies and use LLMs for semantic constraints and quality validation.

Beyond autoregressive and generative adversarial network (GAN)-based methods, recent studies explore diffusion models for traffic generation and detection. Diffusion models learn complex data distributions step by step and show advantages in generation quality and stability. NetDiffus maps traffic time series into two-dimensional representations and performs diffusion modeling in the image space, improving both generation fidelity and downstream task performance [231]. Some studies further combine diffusion models with active learning. NetGuard identifies potential distribution shifts during detection and uses conditional generative models to synthesize additional samples, allowing IDSs to adapt to new attacks with minimal labeled data [232].

As generative modeling becomes mature, research shifts from data generation to reasoning processes and explanation outputs. LLM-based agent architectures change detection from static classification to interactive and interpretable reasoning. These methods use LLMs as the core component and follow a reasoning–action mechanism to dynamically call feature extraction, classification, and knowledge retrieval modules. Some studies further extend this idea to attack mitigation. For example, ShieldGPT detects attacks and generates human-readable explanations and defense suggestions based on security knowledge [233]. These approaches demonstrate the potential of LLM reasoning in improving interpretability and adaptability, but they also introduce high system complexity and resource consumption.

6.3 Capability Boundaries and Cross-Technology

Although LLMs have potential value in security semantic understanding, attack chain association, and explanation of detection results, their direct deployment in real-time detection systems is still clearly limited. Traffic detection in intelligent communication networks usually requires high throughput, low latency, and continuous online processing. However, large-model inference involves high computational overhead, graphics processing unit (GPU) memory consumption, tokenization cost, and context-processing latency. It is difficult to deploy LLMs directly on resource-constrained edge nodes or high-frequency traffic detection links [234]. Therefore, in practical systems, LLMs are more suitable as auxiliary analysis modules in the control plane, management plane, or security operation platform. They can be used for semantic association and explanation generation based on alert results, log texts, threat intelligence, and attack stages. They should not directly replace lightweight real-time detection models on the data plane side [235].

The input representation of LLMs can significantly affect detection latency and stability. Network traffic, logs, and alert information usually need to be serialized, tokenized, and organized into context before being input into LLMs. Input length, context window size, batching strategy, and invocation frequency all affect detection latency [236]. When detection tasks require millisecond-level or sub-second responses, it is not realistic to invoke LLMs for per-flow or per-packet analysis.

In addition, the introduction of LLMs may bring new security risks. LLMs may generate unstable explanations or hallucinated outputs, which may lead to deviations in attack attribution, response recommendations, or risk-level judgment. Prompt injection, adversarial sample construction, sensitive log leakage, and unverifiable model outputs may also affect the trustworthiness of the detection system itself [237–239]. Therefore, LLMs should be used together with KBs, rule verification, trusted auditing, and human review mechanisms. Evidence constraints, result verification, and permission control can reduce the risk of misuse.

The deployment role of LLMs differs across communication scenarios. Cloud platforms can support LLM-based security operation, log summarization, and cross-source alert reasoning because they have stronger computing resources. Edge networks can use small or distilled models for local explanation and rely on cloud-side LLMs for deeper analysis. Satellite networks require delay-tolerant LLM-assisted reasoning on ground stations rather than onboard real-time inference. IoT environments should avoid deploying large models on terminal devices and instead use gateway-side aggregation and cloud-side semantic analysis.

LLMs should also be connected with the other technical routes through clear interfaces. Programmable switches and edge-side detectors can provide compact alerts, flow summaries, and temporal traces to LLMs. KGs can provide structured entities, relations, TTPs, and attack paths as factual constraints. BC can provide trusted evidence indexes, model versions, and audit logs. In return, LLMs can generate explanations, summarize attack chains, suggest response strategies, and assist KG construction.

Overall, LLMs are better suited for upper-layer semantic analysis than for real-time traffic execution. Practical deployment should combine lightweight detectors for online screening with LLMs for alert explanation, threat intelligence analysis, attack-chain reasoning, and decision support. Future LLM-based detection systems should evaluate not only detection accuracy but also token overhead, inference latency, invocation cost, output stability, explanation reliability, privacy protection, and evidence verifiability.

7 Challenges and Future Directions

The preceding sections show that AI-based detection in intelligent communication networks is no longer a single-model classification problem. It has evolved into a system-level problem involving real-time sensing, semantic interpretation, trusted collaboration, and high-level reasoning. Programmable switches, knowledge-driven methods, BC, and LLMs provide complementary capabilities. Programmable switches

improve low-latency traffic sensing and preliminary detection. Knowledge-driven methods improve semantic organization, correlation analysis, and attack attribution. BC improves trusted collaboration, evidence recording, and auditability. LLMs improve contextual understanding, explanation generation, and decision support. However, the four routes also have clear capability boundaries and deployment constraints.

To address the review concern that previous challenges and future directions were too generic, this section reorganizes the discussion from two perspectives. The first perspective compares the capability boundaries of the four technical paradigms. The second perspective refines scenario-specific challenges and future directions for cloud computing, edge computing, satellite networks, and IoT environments. This helps connect the future research problems with the technical routes and application scenarios reviewed in this paper.

7.1 Open Research Problems

Based on the preceding review, several unresolved problems can be further refined from the intersection of the four technical routes and four communication scenarios.

Problem 1: How can real-time traffic sensing be connected with semantic reasoning without violating latency constraints? Programmable switches and lightweight edge detectors can provide low-latency traffic summaries, but their outputs are usually low-level features, counters, or alerts. Knowledge-driven methods and LLMs can provide semantic interpretation, but they introduce additional reasoning latency. Therefore, future studies should design event-triggered and hierarchy-aware interfaces, where only selected high-value alerts, flow summaries, or temporal traces are transferred to KG reasoning and LLM-based explanation modules.

Problem 2: How can knowledge freshness be maintained under dynamic and heterogeneous network environments? Cloud workloads, edge services, satellite topology, and IoT devices change continuously. Static KGs and manually maintained KBs cannot fully reflect new vulnerabilities, attack variants, and service dependencies. Future research should develop incremental KG updating, confidence-aware knowledge fusion, and scenario-specific subgraph synchronization mechanisms. In particular, satellite and edge scenarios require delay-tolerant and lightweight knowledge updating rather than frequent global synchronization.

Problem 3: How can BC record trustworthy evidence without becoming a performance bottleneck? BC improves evidence integrity and auditability, but it cannot record all packet-level or flow-level detection results because of consensus latency, transaction throughput, and storage overhead. Future systems should follow an off-chain detection and on-chain verification principle. Only evidence indexes, model versions, policy updates, knowledge versions, and high-value alerts should be recorded on-chain. A key research problem is how to determine the recording granularity that balances audit completeness, privacy protection, and system overhead.

Problem 4: How can LLM-based detection outputs be constrained and verified? LLMs can summarize alerts, generate explanations, and support attack-chain reasoning, but they may produce hallucinated or unverifiable outputs. This is risky in security detection because incorrect explanations may lead to wrong attribution or inappropriate defense actions. Future LLM-based detection systems should use KGs, rules, detection evidence, and BC audit records as external constraints. Retrieval-augmented generation, evidence citation, human review, and response validation should be integrated into the detection workflow.

Problem 5: How can cross-technology orchestration be evaluated systematically? Most existing studies evaluate a single model or a single system component. However, integrated detection systems require the joint evaluation of latency, detection accuracy, reasoning quality, auditability, privacy leakage, resource

consumption, and operational cost. Future research should develop unified benchmarks and evaluation protocols for multi-technology joint detection, especially for cloud–edge collaboration, satellite–ground collaboration, and massive IoT environments.

7.2 Scenario-Specific Unresolved Problems

In cloud computing, the core problem is scalable and explainable detection under dynamic workloads and multi-tenant coexistence. Programmable switches can reduce traffic collection overhead by performing in-network filtering. However, workload migration and elastic scaling may change traffic baselines. Knowledge-driven methods can correlate alerts with services, assets, and vulnerabilities, but cross-tenant knowledge sharing must avoid privacy leakage. BC can provide audit records for model versions and access behaviors, but ledger growth must be controlled. LLMs can assist security operation by summarizing logs and alerts, but sensitive tenant data must be protected.

In edge computing, the core problem is balancing latency, accuracy, privacy, and resource consumption. Edge nodes can perform local detection, but they cannot support complex reasoning or large-model inference continuously. Knowledge-driven methods should use lightweight local subgraphs and synchronize only key knowledge with the cloud. BC can verify model updates and identities, but consensus mechanisms must be lightweight. LLMs should be invoked selectively for explanation or response support rather than for every detection event.

In satellite networks, the core problem is maintaining detection continuity and trusted collaboration under dynamic topology, long propagation delay, and intermittent connectivity. Programmable detection functions can support lightweight onboard monitoring, but complex analysis should be performed on ground stations or gateways. Knowledge-driven methods should support topology-aware and delay-tolerant attack-chain reconstruction. BC should record key space–ground evidence in batches rather than relying on high-frequency consensus. LLMs can support ground-side reasoning from delayed, sparse, and incomplete evidence.

In IoT environments, the core problem is massive device heterogeneity, scarce labels, and limited terminal resources. Lightweight detection should be performed on devices, gateways, or edge nodes. Knowledge-driven methods should model device types, behavior patterns, malware families, and cross-layer attack relations. BC can support distributed authentication, privacy-preserving learning, and reputation evaluation, but terminal devices cannot bear high communication overhead. LLMs should be deployed at gateways or cloud platforms for event summarization and explanation rather than at constrained endpoints.

7.3 Future Directions

Future AI-based detection in intelligent communication networks should shift from single-model optimization to system-level collaboration. The key is to build a layered and verifiable detection architecture. In this architecture, PSs provide low-latency sensing and preliminary filtering; knowledge-driven modules provide semantic correlation and attribution; BC provides trusted evidence management and accountability; and LLMs provide explanation generation and decision support.

First, future data-plane-based AI detection should focus on lightweight deployability and adaptive collaboration. Since PSs are constrained by pipeline stages, memory size, register operations, and line-rate forwarding requirements, complex AI models cannot be directly deployed in the data plane in most practical scenarios. Future studies should focus on compact detection models, Sketch-based feature extraction, approximate statistics, and rule-assisted inference mechanisms that can be executed within limited switch resources. A feasible engineering path is to establish a closed loop between the data plane and the control

plane. The data plane performs rapid feature extraction and preliminary detection, while the control plane updates detection rules, model parameters, and response policies according to traffic conditions and attack feedback.

Second, future knowledge-driven AI detection should move from static knowledge representation toward dynamic knowledge construction, continuous updating, and closed-loop reasoning. Knowledge should not only serve as external supplementary information or post-hoc explanation. A more effective direction is to integrate KGs into the detection process itself. For example, KGs can constrain feature learning, guide model training, verify detection outputs, and support semantic interpretation of abnormal behaviors. In engineering deployment, lightweight reasoning, incremental reasoning, scenario-specific subgraph extraction, source verification, conflict resolution, and noise filtering should be further explored.

Third, future BC-enabled AI detection should shift from simply recording detection data on-chain to supporting low-cost trusted collaboration and auditable detection processes. High-frequency traffic data and real-time detection results are not suitable for direct on-chain storage because BC systems usually suffer from consensus latency, storage overhead, and limited throughput. A more feasible engineering path is off-chain detection and on-chain verification. AI models perform traffic detection, anomaly analysis, and collaborative inference off-chain, while BC records key evidence indexes, model versions, policy updates, access behaviors, and audit logs on-chain. Smart contracts should formalize evidence submission, detection-result verification, reputation updating, and response accountability, but they should avoid carrying complex detection logic directly.

Fourth, future LLM-based AI detection should be positioned as upper-layer semantic analysis and decision support rather than real-time packet-level or flow-level detection. Real-time detection should still rely mainly on lightweight models, data-plane detection, or edge-side classifiers. LLMs can be deployed in the control plane, management plane, or security operation platform to support log understanding, alert explanation, threat intelligence analysis, attack-chain reasoning, rule generation, and response recommendation. Retrieval-augmented generation, domain-specific fine-tuning, KG constraints, rule-based verification, BC-backed evidence, permission control, and human review should be used to reduce hallucinated outputs and incorrect attribution.

Fifth, future work should establish unified evaluation criteria for multi-technology collaborative detection. Current studies often evaluate accuracy, precision, recall, F1-score, or latency separately, but they rarely evaluate end-to-end detection chains. For integrated systems, evaluation should include detection accuracy, response latency, data-plane overhead, reasoning latency, BC storage cost, consensus delay, LLM invocation cost, explanation correctness, evidence auditability, privacy leakage risk, and cross-scenario transferability. Only with such evaluation can the real value of multi-technology integration be verified.

Overall, future AI-based detection should not pursue the mechanical combination of PSs, KGs, BC, and LLMs. Instead, each technology should be placed at the layer where it has comparative advantages. Programmable switches should be responsible for real-time sensing and preliminary filtering. KGs should be responsible for semantic organization and attack reasoning. BC should be responsible for trusted recording and audit support. LLMs should be responsible for explanation generation and decision support. The unresolved core problem is how to define efficient interfaces, verifiable evidence flows, and measurable end-to-end benefits across these layers.

Acknowledgement: Not applicable.

Funding Statement: This paper is supported by NSFC under Grant No. U2569201 and National Key R&D Program of China under Grant No. 2018YFA0701604.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Jingfu Yan; methodology, Jingfu Yan; software, Jingfu Yan; validation, Jingfu Yan, Huachun Zhou, Xiaojing Fan and Aoran Huang; formal analysis, Jingfu Yan; investigation, Jingfu Yan; resources, Jingfu Yan; data curation, Jingfu Yan; writing—original draft preparation, Jingfu Yan; writing—review and editing, Jingfu Yan, Huachun Zhou, Xiaojing Fan and Aoran Huang; visualization, Jingfu Yan; supervision, Huachun Zhou, Xiaojing Fan and Aoran Huang; project administration, Jingfu Yan; funding acquisition, Huachun Zhou. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

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

References

1. Zhang Z, Ning H, Shi F, Farha F, Xu Y, Xu J, et al. Artificial intelligence in cyber security: research advances, challenges, and opportunities. Artif Intell Rev. 2022;55(2):1029–53. doi:10.1007/s10462-021-09976-0.
2. Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor. 2016;18(2):1153–76. doi:10.1109/COMST.2015.2494502.
3. Ozkan-Okay M, Akin E, Aslan Ö, Kosunalp S, Iliev T, Stoyanov I, et al. A comprehensive survey: evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions. IEEE Access. 2024;12(3):12229–56. doi:10.1109/ACCESS.2024.3355547.
4. Merlano C. Enhancing cyber security through artificial intelligence and machine learning: a literature review. J Cyber Secur. 2024;6(1):89–116. doi:10.32604/jcs.2024.056164.
5. Tian X, Liu J, Qi Q, Li H, Ma X, Zhao C. A collaborative data-plane and control-plane architecture for real-time anomaly detection. IEEE Access. 2025;13:192856–70. doi:10.1109/ACCESS.2025.3631738.
6. Akem AT, Bütün B, Gucciardo M, Fiore M. Jewel: resource-efficient joint packet and flow level inference in programmable switches. In: Proceedings of the IEEE INFOCOM 2024-IEEE Conference on Computer Communications; 2024 May 20–23; Vancouver, BC, Canada. p. 1631–40. doi:10.1109/INFOCOM52122.2024.10621365.
7. Zhao Z, Li Z, Song Z, Zhang F, Chen B. RIDS: towards advanced IDS via RNN model and programmable switches co-designed approaches. In: Proceedings of the IEEE INFOCOM 2024-IEEE Conference on Computer Communications; 2024 May 20–23; Vancouver, BC, Canada. p. 591–600. doi:10.1109/INFOCOM52122.2024.10621290.
8. Hu B, Zhou C, Tian YC, Qin Y, Junping X. A collaborative intrusion detection approach using blockchain for multimicrogrid systems. IEEE Trans Syst Man Cybern Syst. 2019;49(8):1720–30. doi:10.1109/TSMC.2019.2911548.
9. Rahman MA, Hossain MS, Islam MS, Alrajeh NA, Muhammad G. Secure and provenance enhanced Internet of health things framework: a blockchain managed federated learning approach. IEEE Access. 2020;8:205071–87. doi:10.1109/ACCESS.2020.3037474.
10. Yang H, Zhan K, Kadoch M, Liang Y, Cheriet M. BLCS: brain-like distributed control security in cyber physical systems. IEEE Netw. 2020;34(3):8–15. doi:10.1109/MNET.011.1900275.
11. Suri N. Distributed systems security knowledge area version. Cyber Secur Body Knowl. 2021.
12. Kumar P, Javeed D, Kumar R, Islam AKMN. Blockchain and explainable AI for enhanced decision making in cyber threat detection. Softw Pract Exp. 2024;54(8):1337–60. doi:10.1002/spe.3319.
13. Goundar S. Blockchain-AI integration for resilient real-time cyber security. In: Proceeding of the 2024 Global Congress on Emerging Technologies (GCET-2024); 2024 Dec 9–11; Gran Canaria, Spain. p. 342–9. doi:10.1109/GCET64327.2024.10934609.
14. Xu H, Wang S, Li N, Wang K, Zhao Y, Chen K, et al. Large language models for cyber security: a systematic literature review. ACM Trans Softw Eng Methodol. 2025:3769676. doi:10.1145/3769676.
15. Song S, Lin Y, Guo B, Di Q, Lv R. Scalable distributed semantic network for knowledge management in cyber physical system. J Parallel Distrib Comput. 2018;118:22–33. doi:10.1016/j.jpdc.2017.11.014.

16. Nosouhi MR, Yu S, Zhou W, Grobler M, Keshtiar H. Blockchain for secure location verification. J Parallel Distrib Comput. 2020;136:40–51. doi:10.1016/j.jpdc.2019.10.007.
17. Saqib M, Elbiaze H, Glitho RH. A data-driven approach to mitigate evolving volumetric attacks in programmable networks. IEEE Trans Mach Learn Commun Netw. 2025;3:883–902. doi:10.1109/TMLCN.2025.3594659.
18. Mustafa Z, Amin R, Aldabbas H, Ahmed N. Intrusion detection systems for software-defined networks: a comprehensive study on machine learning-based techniques. Clust Comput. 2024;27(7):9635–61. doi:10.1007/s10586-024-04430-6.
19. Zhao X, Jiang R, Han Y, Li A, Peng Z. A survey on cybersecurity knowledge graph construction. Comput Secur. 2024;136(11):103524. doi:10.1016/j.cose.2023.103524.
20. Kumari A, Kumar S. Mitigating intrusion attacks in VANETs through reliable communication with graph-based deep learning approaches. J Supercomput. 2025;81(15):1467. doi:10.1007/s11227-025-07945-w.
21. Sarker IH. Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: a comprehensive overview. Security Priv. 2023;6(5):e295. doi:10.1002/spy2.295.
22. Yang X, Peng G, Liu S, Zhang D, Li C, Liu X, et al. A survey on intelligent detection for APT attacks. China Commun. 2025;22(11):103–31. doi:10.23919/jcc.fa.2023-0667.202511.
23. Alkadi O, Moustafa N, Turnbull B, Choo KR. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. IEEE Internet Things J. 2021;8(12):9463–72. doi:10.1109/JIOT.2020.2996590.
24. Tsang YP, Wu CH, Dong N. A federated-ANFIS for collaborative intrusion detection in securing decentralized autonomous organizations. IEEE Trans Eng Manag. 2024;71(23):12529–41. doi:10.1109/TEM.2023.3304409.
25. Arazzi M, Nicolazzo S, Nocera A. A fully privacy-preserving solution for anomaly detection in IoT using federated learning and homomorphic encryption. Inf Syst Front. 2025;27(1):367–90. doi:10.1007/s10796-023-10443-0.
26. Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. In: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics; 2020 Aug 26–28; Online. p. 2938–48. doi:10.48550/arxiv.1807.00459.
27. Khraisat A, Alazab A, Singh S, Jan T, Jr Gomez A. Survey on federated learning for intrusion detection system: concept, architectures, aggregation strategies, challenges, and future directions. ACM Comput Surv. 2025;57(1):1–38. doi:10.1145/3687124.
28. Oun A, Wince K, Cheng X. The role of artificial intelligence in boosting cybersecurity and trusted embedded systems performance: a systematic review on current and future trends. IEEE Access. 2025;13:55258–76. doi:10.1109/access.2025.3554739.
29. Bui MT, Boffa M, Valentim RV, Navarro JM, Chen F, Bao X, et al. A systematic comparison of large language models performance for intrusion detection. Proc ACM Netw. 2024;2(CoNEXT4):1–23. doi:10.1145/3696379.
30. Bosshart P, Daly D, Gibb G, Izzard M, McKeown N, Rexford J, et al. P4: programming protocol-independent packet processors. SIGCOMM Comput Commun Rev. 2014;44(3):87–95. doi:10.1145/2656877.2656890.
31. Liu Y, Yu FR, Li X, Ji H, Leung VCM. Blockchain and machine learning for communications and networking systems. IEEE Commun Surv Tutor. 2020;22(2):1392–431. doi:10.1109/COMST.2020.2975911.
32. Kfoury EF, Crichigno J, Bou-Harb E. An exhaustive survey on P4 programmable data plane switches: taxonomy, applications, challenges, and future trends. IEEE Access. 2021;9:87094–155. doi:10.1109/ACCESS.2021.3086704.
33. Hauser F, Häberle M, Merling D, Lindner S, Gurevich V, Zeiger F, et al. A survey on data plane programming with P4: fundamentals, advances, and applied research. J Netw Comput Appl. 2023;212(3):103561. doi:10.1016/j.jnca.2022.103561.
34. Yan Z, Liu J. A review on application of knowledge graph in cybersecurity. In: 2020 International Signal Processing, Communications and Engineering Management Conference (ISPCEM); 2020 Nov 27–29; Montreal, QC, Canada. p. 240–3. doi:10.1109/ispcem52197.2020.00055.
35. Zuo Y, Guo J, Gao N, Zhu Y, Jin S, Li X. A survey of blockchain and artificial intelligence for 6G wireless communications. IEEE Commun Surv Tutor. 2023;25(4):2494–528. doi:10.1109/COMST.2023.3315374.
36. Ali S, Li Q, Yousafzai A. Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: a survey. Ad Hoc Netw. 2024;152(6):103320. doi:10.1016/j.adhoc.2023.103320.

37. Zhang J, Bu H, Wen H, Liu Y, Fei H, Xi R, et al. When LLMs meet cybersecurity: a systematic literature review. Cybersecurity. 2025;8(1):55. doi:10.1186/s42400-025-00361-w.
38. Chen X, Wu C, Liu X, Huang Q, Zhang D, Zhou H, et al. Empowering network security with programmable switches: a comprehensive survey. IEEE Commun Surv Tutor. 2023;25(3):1653–704. doi:10.1109/COMST.2023.3265984.
39. Heidari A, Ali Jabraeil Jamali M. Internet of Things intrusion detection systems: a comprehensive review and future directions. Clust Comput. 2023;26(6):3753–80. doi:10.1007/s10586-022-03776-z.
40. AboulEla SG, Kashef RF. Leveraging large language models, graph neural networks, and explainable AI for revolutionizing the next-generation network intrusion detection systems. J Intell Inf Syst. 2025;63(5):1807–35. doi:10.1007/s10844-025-00964-2.
41. Modi C, Patel D, Borisaniya B, Patel H, Patel A, Rajarajan M. A survey of intrusion detection techniques in Cloud. J Netw Comput Appl. 2013;36(1):42–57. doi:10.1016/j.jnca.2012.05.003.
42. Zhuo M, Liu L, Zhou S, Tian Z. Survey on security issues of routing and anomaly detection for space information networks. Sci Rep. 2021;11(1):22261. doi:10.1038/s41598-021-01638-z.
43. Nidadavolu K, Somasekhar G. Enhancing cloud security: a comprehensive review of deep learning algorithms. In: Proceedings of the 2025 6th International Conference on Recent Advances in Information Technology (RAIT); 2025 Mar 6–8; Dhanbad, India. p. 1–6. doi:10.1109/RAIT65068.2025.11089291.
44. Al-Shurbaji T, Anbar M, Manickam S, Hasbullah IH, Alfriehat N, Ahmad Alabsi B, et al. Deep learning-based intrusion detection system for detecting IoT botnet attacks: a review. IEEE Access. 2025;13(8):11792–822. doi:10.1109/ACCESS.2025.3526711.
45. Zhang T, Kong F, Deng D, Tang X, Wu X, Xu C, et al. Moving target defense meets artificial-intelligence-driven network: a comprehensive survey. IEEE Internet Things J. 2025;12(10):13384–97. doi:10.1109/JIOT.2025.3533016.
46. Tan Y, Koketsu Rodrigues T, Kato N, Ariyoshi M, Hasegawa Y. Pivotal AI paradigms for satellite communication security: from machine learning to large language models. IEEE Commun Surv Tutor. 2026;28(1):4654–89. doi:10.1109/COMST.2026.3658670.
47. Popoola SI, Tsado Y, Ogunjinmi AA, Sanchez-Velazquez E, Peng Y, Rawat DB. Multi-stage deep learning for intrusion detection in industrial Internet of Things. IEEE Access. 2025;13(3):60532–55. doi:10.1109/ACCESS.2025.3557959.
48. Keegan N, Ji SY, Chaudhary A, Concolato C, Yu B, Jeong DH. A survey of cloud-based network intrusion detection analysis. Hum Centric Comput Inf Sci. 2016;6(1):19. doi:10.1186/s13673-016-0076-z.
49. Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv. 2009;41(3):1–58. doi:10.1145/1541880.1541882.
50. Sommer R, Paxson V. Outside the closed world: on using machine learning for network intrusion detection. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy; 2010 May 16–19; Oakland, CA, USA. p. 305–16. doi:10.1109/SP.2010.25.
51. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. Comput Secur. 2019;86(1):147–67. doi:10.1016/j.cose.2019.06.005.
52. Usama M, Qadir J, Raza A, Arif H, Yau KA, Elkhatib Y, et al. Unsupervised machine learning for networking: techniques, applications and research challenges. IEEE Access. 2019;7:65579–615. doi:10.1109/ACCESS.2019.2916648.
53. Moghaddasi K, Rajabi S, Hosseinzadeh M. Intrusion detection systems for enhanced security in mobile edge computing: a systematic review and survey of the applications, challenges, and future directions. Wirel Pers Commun. 2025;145(1):113–75. doi:10.1007/s11277-025-11846-9.
54. Rocha A, Monteiro M, Mattos C, Dias M, Soares J, Magalhães R, et al. Edge AI for Internet of medical things: a literature review. Comput Electr Eng. 2024;116(8):109202. doi:10.1016/j.compeleceng.2024.109202.
55. Assistant MT, Sivanesh C, Sanjeev Kumar M, Dharineesh S. Lightweight edge computing framework for IoT botnet detection in dynamic environment. In: Proceedings of the 2025 Third International Conference on Emerging Applications of Material Science and Technology (ICEAMST); 2025 Nov 3–5; Bengaluru, India. p. 1154–9. doi:10.1109/ICEAMST67459.2025.11335881.

56. Golec M, Gill SS, Bahsoon R, Rana O. BioSec: a biometric authentication framework for secure and private communication among edge devices in IoT and industry 4.0. IEEE Consum Electron Mag. 2022;11(2):51–6. doi:10.1109/MCE.2020.3038040.
57. Gutti C, Thumula K, Balbudhe P. Federated learning for distributed IoT security: a privacy-preserving approach to intrusion detection. IEEE Access. 2025;13:135863–75. doi:10.1109/ACCESS.2025.3592481.
58. Vyas A, Lin PC, Hwang RH, Tripathi M. Privacy-preserving federated learning for intrusion detection in IoT environments: a survey. IEEE Access. 2024;12:127018–50. doi:10.1109/ACCESS.2024.3454211.
59. Zhang W, Zeadally S, Li W, Zhang H, Hou J, Leung VCM. Edge AI as a service: configurable model deployment and delay-energy optimization with result quality constraints. IEEE Trans Cloud Comput. 2023;11(2):1954–69. doi:10.1109/TCC.2022.3175725.
60. Borra CR, Rayala RV, Pareek PK, Cheekati S. Optimizing security in satellite-integrated IoT networks: a hybrid deep learning approach for intrusion detection with JBOA and NOA. In: Proceedings of the 2025 International Conference on Intelligent and Cloud Computing; 2025 May 2–3; Bhubaneswar, India. p. 1–8. doi:10.1109/ICoICC64033.2025.11052096.
61. Ahmad SZ, Qamar F. A hybrid AI based framework for enhancing security in satellite based IoT networks using high performance computing architecture. Sci Rep. 2024;14(1):30695. doi:10.1038/s41598-024-78262-0.
62. De S, Bermudez-Edo M, Xu H, Cai Z. Deep generative models in the industrial Internet of Things: a survey. IEEE Trans Ind Inform. 2022;18(9):5728–37. doi:10.1109/TII.2022.3155656.
63. Frustaci M, Pace P, Aloi G, Fortino G. Evaluating critical security issues of the IoT world: present and future challenges. IEEE Internet Things J. 2018;5(4):2483–95. doi:10.1109/JIOT.2017.2767291.
64. Yan Q, Huang W, Luo X, Gong Q, Yu FR. A multi-level DDoS mitigation framework for the industrial Internet of Things. IEEE Commun Mag. 2018;56(2):30–6. doi:10.1109/MCOM.2018.1700621.
65. Mallidi SKR, Ramisetty RR. Advancements in training and deployment strategies for AI-based intrusion detection systems in IoT: a systematic literature review. Discov Internet Things. 2025;5(1):8. doi:10.1007/s43926-025-00099-4.
66. Victor P, Lashkari AH, Lu R, Sasi T, Xiong P, Iqbal S. IoT malware: an attribute-based taxonomy, detection mechanisms and challenges. Peer Peer Netw Appl. 2023;16(3):1380–431. doi:10.1007/s12083-023-01478-w.
67. Biggio B, Roli F. Wild patterns: ten years after the rise of adversarial machine learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security; 2018 Oct 15–19; Toronto, ON, Canada.
68. Xiao L, Wan X, Lu X, Zhang Y, Wu D. IoT security techniques based on machine learning: how do IoT devices use AI to enhance security? IEEE Signal Process Mag. 2018;35(5):41–9. doi:10.1109/MSP.2018.2825478.
69. Sa'ad IA, Udanor CN, Ezema ME, Markus C, Adaji MA. Deep learning and explainable AI models for intrusion detection in space-ground communication networks: a review. Inf Technol. 2026;9(1):64–75.
70. Badiginchala HP. Advanced artificial intelligence-based cybersecurity intrusion identification system for cloud computing environments. In: Proceedings of the 2025 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics; 2025 Dec 1–2; Lombok, Indonesia. p. 1–7. doi:10.1109/ICoABCD67551.2025.11470882.
71. Li X, Zhang T, Wang J, Han Z, Wang N, Fan S, et al. Protect NTN-IoT security by malicious traffic detection: a multidimensional hypergraph learning approach. IEEE Internet Things J. 2026;13(3):3594–607. doi:10.1109/JIOT.2025.3582027.
72. Soursos S, Žarko IP, Zwickl P, Gojmerac I, Bianchi G, Carrozzo G. Towards the cross-domain interoperability of IoT platforms. In: Proceedings of the 2016 European conference on networks and communications (EuCNC); 2016 Jun 27–30; Athens, Greece. p. 398–402. doi:10.1109/EuCNC.2016.7561070.
73. Park J, Eom T, Kim H, Park H, Yoon Z, Park J. Threat vector–hierarchical attack representation model-based threat modeling and security assessment for satellite networks. Appl Sci. 2025;15(5):2751. doi:10.3390/app15052751.
74. Liu FT, Ting KM, Zhou ZH. Isolation forest. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining; 2008 Dec 15–19; Pisa, Italy. p. 413–22. doi:10.1109/ICDM.2008.17.

75. Gu Y, Zhang L, Zhang Y, Du Y. SatFedGuard: semi-supervised federated contrastive learning with RL-assisted bidirectional distillation for anomaly traffic detection in satellite networks. IEEE Trans Netw Serv Manag. 2026;23(22):4382–402. doi:10.1109/TNSM.2026.3685416.
76. Wang Y, Su Z, Ni J, Zhang N, Shen X. Blockchain-empowered space-air-ground integrated networks: opportunities, challenges, and solutions. IEEE Commun Surv Tutor. 2022;24(1):160–209. doi:10.1109/COMST.2021.3131711.
77. Xu D, Liang Y, Li C, Zhou X, Zhang C, Zhu L, et al. LMT-SDNN: a lightweight malicious traffic detection method for the Internet of Things based on multiteacher distillation. IEEE Internet Things J. 2026;13(8):15665–77. doi:10.1109/JIOT.2026.3657416.
78. Kianpisheh S, Taleb T. A survey on in-network computing: programmable data plane and technology specific applications. IEEE Commun Surv Tutor. 2023;25(1):701–61. doi:10.1109/COMST.2022.3213237.
79. Gray N, Dietz K, Seufert M, Hossfeld T. High performance network metadata extraction using P4 for ML-based intrusion detection systems. In: Proceedings of the 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR); 2021 Jun 7–10; Paris, France. p. 1–7. doi:10.1109/HPSR52026.2021.9481849.
80. Mittal S, Kotha H, Anand Krishna M, Tammana P. AdaFlow: efficient in-network traffic classification using programmable switches. In: Proceedings of the 2024 IFIP Networking Conference (IFIP Networking); 2024 Jun 3–6; Thessaloniki, Greece. p. 258–66.
81. Mittal S, Tammana P. Efficient in-network traffic classification using programmable switches with AdaFlow. IEEE Trans Netw Serv Manag. 2025;22(6):5532–49. doi:10.1109/TNSM.2025.3607406.
82. Yang Z, Zhang X, Wang X, Ren J, Lin R, Wang S, et al. Mix sketch: differentiated and accurate per-flow measurement for programmable networks. In: Proceedings of the GLOBECOM 2025–2025 IEEE Global Communications Conference; 2025 Dec 8–12; Taipei, Taiwan. p. 5562–7. doi:10.1109/GLOBECOM59602.2025.11431690.
83. Huang Y, Chen L, Peng Z, Cui L. SP-sketch: persistent flow detection with sliding windows on programmable switches. Comput Mater Contin. 2025;84(3):6015–34. doi:10.32604/cmc.2025.066717.
84. Akem AT, Bütün B, Gucciardo M, Fiore M. Practical and general-purpose flow-level inference with random forests in programmable switches. IEEE Trans Netw. 2025;33(5):2489–506. doi:10.1109/TON.2025.3564465.
85. Lee JH, Singh K. SwitchTree: in-network computing and traffic analyses with Random Forests. Neural Comput Appl. 2025;37(28):23143–54. doi:10.1007/s00521-020-05440-2.
86. He Z, Qu Y, Jin D. Real-time power system event detection on programmable network switches with synchrophasor data. In: Proceedings of the 2025 IEEE International Conference on Communications; 2025 Jun 8–12; Montreal, QC, Canada. p. 3918–23. doi:10.1109/ICC52391.2025.11161366.
87. Li Q, Lin J, Xie G, Guan Z, Luan Z, Qi Z, et al. Distributed multi-task in-network classification on programmable switches by ensemble models. IEEE Trans Netw. 2025;33(6):3302–17. doi:10.1109/TON.2025.3590275.
88. Razavi K, Fard SD, Karlos G, Nigade V, Mühlhüser M, Wang L. NetNN: neural intrusion detection system in programmable networks. In: Proceedings of the 2024 IEEE Symposium on Computers and Communications (ISCC); 2024 Jun 26–29; Paris, France: IEEE. p. 1–8. doi:10.1109/ISCC61673.2024.10733725.
89. Xie G, Li Q, Duan G, Lin J, Dong Y, Jiang Y, et al. Empowering in-network classification in programmable switches by binary decision tree and knowledge distillation. IEEE/ACM Trans Netw. 2024;32(1):382–95. doi:10.1109/TNET.2023.3287091.
90. Xie G, Li Q, Dong Y, Duan G, Jiang Y, Duan J. Mousika: enable general in-network intelligence in programmable switches by knowledge distillation. In: Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications; 2022 May 2–5; London, UK. p. 1938–47. doi:10.1109/INFOCOM48880.2022.9796936.
91. Akem AT, Gucciardo M, Fiore M. Flowrest: practical flow-level inference in programmable switches with random forests. In: Proceedings of the IEEE INFOCOM 2023-IEEE Conference on Computer Communications; 2023 May 17–20; New York, NY, USA. p. 1–10. doi:10.1109/INFOCOM53939.2023.10229100.
92. Yan J, Zhou H. SMV-sketch: a sketch for simultaneous detection of DDoS and heavy flows. In: Proceedings of the 2025 3rd International Conference on Intelligent Communication and Networking (ICN); 2025 Nov 8–10; Lhasa, China. p. 25–8. doi:10.1109/ICN68016.2025.11384089.

93. Tang L, Huang Q, Lee PPC. MV-sketch: a fast and compact invertible sketch for heavy flow detection in network data streams. In: Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications; 2019 Apr 29–May 2; Paris, France. p. 2026–34. doi:10.1109/infocom.2019.8737499.
94. Clemens V, Schulz LC, Gartner M, Hausheer D. DDoS detection in P4 using HYPERLOGLOG and COUNTMIN sketches. In: Proceedings of the NOMS 2023–2023 IEEE/IFIP Network Operations and Management Symposium; 2023 May 8–12; Miami, FL, USA. p. 1–6. doi:10.1109/NOMS56928.2023.10154315.
95. Xie G, Li Q, Cui C, Li R, Ma L, Qi Z, et al. Intelligent in-network attack detection on programmable switches with Soterv2. IEEE Trans Dependable Secure Comput. 2025;22(1):440–56. doi:10.1109/TDSC.2024.3402955.
96. Yan J, Xu H, Liu Z, Li Q, Xu K, Xu M, et al. Brain-on-switch: towards advanced intelligent network data plane via NN-driven traffic analysis at line-speed. In: Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24); 2024 Apr 16–18; Santa Clara, CA, USA. p. 419–40.
97. He Z, Qu Y, Chen G, Raj RS, Lin H, Jin D. Towards secure and resilient synchrophasor networks using P4 programmable switches. In: Proceedings of the 2024 IEEE Green Technologies Conference (GreenTech); 2024 Apr 3–5; Springdale, AR, USA. p. 17–21. doi:10.1109/GreenTech58819.2024.10520393.
98. Luo J, Zhou Z, Tang Q, Chen R, Chen X, Wang D, et al. P4-IDet: a programmable switch-based framework for real-time and high-accuracy traffic anomaly detection in ICPSs. In: Proceedings of the IECON 2025–51st Annual Conference of the IEEE Industrial Electronics Society; 2025 Oct 14–17; Madrid, Spain. p. 1–6. doi:10.1109/IECON58223.2025.11221702.
99. Zang M, Zaballa EO, Dittmann L, Wu J. Dynamic adaptation of in-band network monitoring via meta learning. In: Proceedings of the 2025 IEEE 50th Conference on Local Computer Networks (LCN); 2025 Oct 13–16; Sydney, Australia. p. 1–8. doi:10.1109/LCN65610.2025.11146389.
100. Li Y, Chen T, Ciric V, Wang C. AMSO-INT: reinforcement learning-driven dynamic adaptive in-band telemetry. In: Proceedings of the 2nd Workshop on Networks for AI Computing. Coimbra Portugal. New York, NY, USA: ACM; 2025. p. 67–73. doi:10.1145/3748273.3749207.
101. Ríos-Guiral S, Lahmadi A, Botero JF, Gutiérrez SA. Leveraging reinforcement learning for traffic engineering in programmable networks: a survey. IEEE Commun Surv Tutor. 2026;28:1318–50. doi:10.1109/COMST.2025.3632870.
102. Quan W, Xu Z, Liu M, Cheng N, Liu G, Gao D, et al. AI-driven packet forwarding with programmable data plane: a survey. IEEE Commun Surv Tutor. 2023;25(1):762–90. doi:10.1109/COMST.2022.3217613.
103. Hu Z, Lin H, Qu Y, Jin D. Leveraging compact data accumulator to enable in-network anomaly detection in programmable switches for power grids. In: Proceedings of the 2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm); 2024 Sep 17–20; Oslo, Norway. p. 548–54. doi:10.1109/SmartGridComm60555.2024.10738074.
104. Rajagopalan S, Sivalingam KM, Shami G, Lyonnais M. ALADA: packet-level analysis for line rate anomaly detection using a programmable dataplane switch. IEEE Open J Commun Soc. 2026;7:4200–22. doi:10.1109/OJCOMS.2026.3686225.
105. Misa C. Traffic monitoring using programmable switch hardware for in-network aggregation. Eugene, OR, USA: Department of Computer and Information Science, University of Oregon; 2023. Report No.: AREA-202303.
106. Li R, Li Q, Lin T, Zou Q, Zhao D, Huang Y, et al. DeviceRadar: online IoT device fingerprinting in ISPs using programmable switches. IEEE/ACM Trans Netw. 2024;32(5):3854–69. doi:10.1109/TNET.2024.3398778.
107. Yan J, Zhou H, Wang W. DPZTN: data-plane-based access control zero-trust network. Comput Syst Sci Eng. 2025;49(1):499–531.
108. Zhong L, Wu J, Li Q, Peng H, Wu X. A comprehensive survey on automatic knowledge graph construction. ACM Comput Surv. 2024;56(4):1–62. doi:10.1145/3618295.
109. Putra MAR, Ahmad T, Hostiadi DP. B-CAT: a model for detecting botnet attacks using deep attack behavior analysis on network traffic flows. J Big Data. 2024;11(1):49. doi:10.1186/s40537-024-00900-1.
110. Byrapuneni LP, Saidireddy M. An efficient cluster based multi-label classification model for advanced persistent threat attacks detecting. Int J Saf Secur Eng. 2024;14(2):541–51. doi:10.18280/ijsse.140221.

111. Rajesh P, Alam M, Tahernezhadi M, Monika A, Chanakya G. Analysis of cyber threat detection and emulation using MITRE attack framework. In: Proceedings of the 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA); 2022 Sep 5–7; San Antonio, TX, USA. p. 4–12. doi:10.1109/IDSTA55301.2022.9923170.
112. Qiu X, Lv M, Chen T, Zhu T, Song Q, Zhu Z. Zoomer: an APT TTP recognition system *via* deep & wide provenance graph learning. IEEE Trans Dependable Secure Comput. 2026;23(3):4893–909. doi:10.1109/TDSC.2025.3646355.
113. Guo Q, Li M, Wang W, Liu Y. A dynamic deployment method of security services based on malicious behavior knowledge base. Sensors. 2022;22(22):9021. doi:10.3390/s22229021.
114. Feng K, Zhou H, Wang W, Yan J, Fan X. Construction of a cybersecurity behavior knowledge base for malicious behavior analysis. ReBICTE. 2024;10:144–68. doi:10.64799/rebicte.v10.10.
115. Wang W, Zhou H, Yan J, Fan X. Knowledge-driven rapid adaptation to new attacks. IEEE Trans Cogn Commun Netw. 2025;11(3):1996–2012. doi:10.1109/TCCN.2024.3464489.
116. Wang Z, Fei S, Hu Y, Shan D, Xiao S, You L, et al. Automated attack knowledge graph construction with large language models. In: Proceedings of the 2025 2nd International Conference on Computer and Multimedia Technology; 2025 Jun 6–8; Sanming, China. p. 700–6. doi:10.1145/3757749.3757864.
117. Zhang Y, Zhao X, Ma Y, Ma H, Guan Y, Yang G, et al. MM-AttacKG: a multimodal approach to attack graph construction with large language models. Knowl Based Syst. 2026;338(1):115483. doi:10.1016/j.knosys.2026.115483.
118. Yang X, Zhong R, Chen Y, Peng G, Yao D, Chen C, et al. CTI-Thinker: an LLM-driven system for CTI knowledge graph construction and attack reasoning. Cybersecurity. 2026;9(1):106. doi:10.1186/s42400-025-00505-y.
119. Cheng Y, Bajaber O, Tsegai SA, Song D, Gao P. CTINexus: automatic cyber threat intelligence knowledge graph construction using large language models. In: Proceedings of the 2025 IEEE 10th European Symposium on Security and Privacy (EuroS&P); 2025 Jun 30–Jul 4; Venice, Italy. p. 923–38. doi:10.1109/EuroSP63326.2025.00057.
120. Tailhardat L, Eurecom, Troncy R, Chabot Y. Anomaly detection using knowledge graphs: a survey for network management and cybersecurity application. ACM Comput Surv. 2026;102:3830090. doi:10.1145/3830090.
121. Ge X, Chen Z, Yan L, Yang Y. Digital intelligent world: from data-driven AI to knowledge-enabled intelligent agents. IEEE Trans Knowl Data Eng. 2026;38(3):1604–21. doi:10.1109/TKDE.2026.3651336.
122. Zhang S, Yin H, Chen T, Hung QVN, Huang Z, Cui L. GCN-based user representation learning for unifying robust recommendation and fraudster detection. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval; 2020 Jul 25–30; Virtual Event. p. 689–98. doi:10.1145/3397271.3401165.
123. Nandiya P, Mohsin A, Ibrahim A, Sarker IH, Janicke H. Bridg-ics: AI-grounded knowledge graphs for intelligent threat analytics in industry 5.0 cyber-physical systems. Cybersecurity. 2026;9(1):167. doi:10.1186/s42400-026-00597-0.
124. Legg PA, Buckley O, Goldsmith M, Creese S. Automated insider threat detection system using user and role-based profile assessment. IEEE Syst J. 2017;11(2):503–12. doi:10.1109/JSYST.2015.2438442.
125. Kim J, Park M, Kim H, Cho S, Kang P. Insider threat detection based on user behavior modeling and anomaly detection algorithms. Appl Sci. 2019;9(19):4018. doi:10.3390/app9194018.
126. Sharma B, Pokharel P, Joshi B. User behavior analytics for anomaly detection using LSTM autoencoder-insider threat detection. In: Proceedings of the 11th International Conference on Advances in Information Technology; 2020 Jul 1–3; Bangkok Thailand. p. 1–9. doi:10.1145/3406601.3406610.
127. Ye X, Han MM. An improved feature extraction algorithm for insider threat using hidden Markov model on user behavior detection. Inf Comput Secur. 2022;30(1):19–36. doi:10.1108/ics-12-2019-0142.
128. Yamauchi M, Ohsita Y, Murata M, Ueda K, Kato Y. Anomaly detection for smart home based on user behavior. In: Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE); 2019 Jan 11–13; Las Vegas, NV, USA. p. 1–6. doi:10.1109/icce.2019.8661976.
129. Böse B, Avasarala B, Tirthapura S, Chung YY, Steiner D. Detecting insider threats using RADISH: a system for real-time anomaly detection in heterogeneous data streams. IEEE Syst J. 2017;11(2):471–82. doi:10.1109/JSYST.2016.2558507.

130. Zhao Z, Li Z, Song Z, Li W, Zhang F. Trident: a universal framework for fine-grained and class-incremental unknown traffic detection. In: Proceedings of the ACM Web Conference 2024; 2024 May 13–17; Singapore. p. 1608–19. doi:10.1145/3589334.3645407.
131. Chen L, Gao S, Liu B. An improved density peaks clustering algorithm based on grid screening and mutual neighborhood degree for network anomaly detection. Sci Rep. 2022;12(1):1409. doi:10.1038/s41598-021-02038-z.
132. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems. 2020;2:429–50. doi:10.48550/arxiv.1812.06127.
133. Huang Y, Chu L, Zhou Z, Wang L, Liu J, Pei J, et al. Personalized cross-*Silo* federated learning on non-IID data. Proc AAAI Conf Artif Intell. 2021;35(9):7865–73. doi:10.1609/aaai.v35i9.16960.
134. Meng L, Wei Y, Pan R, Zhou S, Zhang J, Chen W. VADAF: visualization for abnormal client detection and analysis in federated learning. ACM Trans Interact Intell Syst. 2021;11(3–4):1–23. doi:10.1145/3426866.
135. Savage S, Wetherall D, Karlin A, Anderson T. Practical network support for IP traceback. In: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication; 2000 Aug 28–Sep 1; Stockholm Sweden. p. 295–306. doi:10.1145/347059.347560.
136. Sun S, Huang C, Wu T, Shen Y. SecTKG: a knowledge graph for open-source security tools. Int J Intell Syst. 2023;2023(1):4464974. doi:10.1155/2023/4464974.
137. Nur AY. Efficient probabilistic packet marking for AS traceback. In: Proceedings of the 2021 International Symposium on Networks, Computers and Communications (ISNCC); 2021 Oct 31–Nov 2; Dubai, United Arab Emirates. p. 1–6. doi:10.1109/isncc52172.2021.9615716.
138. Murugesan V, Selvaraj MS, Yang MH. HPSIPT: a high-precision single-packet IP traceback scheme. Comput Netw. 2018;143(2):275–88. doi:10.1016/j.comnet.2018.07.013.
139. Kamaldeep, Malik M, Dutta M. Implementation of single-packet hybrid IP traceback for IPv4 and IPv6 networks. IET Inf Secur. 2018;12(1):1–6. doi:10.1049/iet-ifs.2015.0483.
140. Jeong E, Lee B. An IP traceback protocol using a compressed hash table, a sinkhole router and data mining based on network forensics against network attacks. Future Gener Comput Syst. 2014;33(6):42–52. doi:10.1016/j.future.2013.10.023.
141. Agarwal K, Rozner E, Dixon C, Carter J. SDN traceroute: tracing SDN forwarding without changing network behavior. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking; 2014 Aug 22; Chicago, IL, USA. p. 145–50. doi:10.1145/2620728.2620756.
142. Castillo-Fernández E, Díaz-Verdejo J, Estepa AR, Estepa AA, Muñoz CJ, Mabinabeitia G. Multistep cyberattacks detection using a flexible multilevel system for alerts and events correlation. In: Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference; 2023 Jun 14–15; Stavanger, Norway. p. 1–6. doi:10.1145/3590777.3590778.
143. Muñoz-Calle J, Alonso RE, Estepa Alonso A, Díaz-Verdejo JE, Castillo Fernández E, Madinabeitia G. A flexible multilevel system for mitre ATT&CK model-driven alerts and events correlation in cyberattacks detection. Jucs. 2024;30(9):1184–204. doi:10.3897/jucs.131686.
144. Chen X, Qian J, Shen S, Zheng S. Traceback of attack chains in an intelligent power grid ATT&CK framework based on multi-labels. In: Proceedings of the International Conference on Mathematics and Machine Learning; 2023 Nov 24–26; Nanjing, China. p. 292–7. doi:10.1145/3653724.3653775.
145. Bijalwan A, Sando S, Lemma M. An anatomy for recognizing network attack intention. Int J Recent Technol Eng IJRTE. 2019;8(3):803–16. doi:10.35940/ijrte.c4022.098319.
146. Wang J, Zeng Y, Leng T, Zhao J, Huang C. From soup to nuts: a hierarchical relation-based IP attribution approach for cyber threat traceback. In: Proceedings of the 2025 IEEE International Conference on High Performance Computing and Communications (HPCC); 2025 Aug 13–15; Exeter, UK. p. 801–8. doi:10.1109/HPCC67675.2025.00120.
147. Xiao F, Chen E, Xu Q, Zhang X. ICSTrace: a malicious IP traceback model for attacking data of the industrial control system. Secur Commun Netw. 2021;2021(4):7525092. doi:10.1155/2021/7525092.
148. Hadem P, Saikia DK, Moulik S. An SDN-based intrusion detection system using SVM with selective logging for IP traceback. Comput Netw. 2021;191(4):108015. doi:10.1016/j.comnet.2021.108015.

149. Puchalski D, Pawlicki M, Kozik R, Renk R, Choraś M. Trustworthy AI-based cyber-attack detector for network cyber crime forensics. In: Proceedings of the 19th International Conference on Availability, Reliability and Security; 2024 Jul 30–Aug 2; Vienna, Austria. p. 1–8. doi:10.1145/3664476.3670880.
150. Toluwaleke O, Uno D, Olagbaju IE, Fagbohun O, Ayodele PE, Effiong Offiong E, et al. Temporal-XAI: an attention-based explainable framework for DoS attack detection with temporal pattern analysis. In: Proceedings of the 2025 10th International Conference on Machine Learning Technologies (ICMLT); 2025 May 23–25; Helsinki, Finland. p. 439–50. doi:10.1109/ICMLT65785.2025.11193310.
151. Li L, Fan Y, Tse M, Lin KY. A review of applications in federated learning. Comput Ind Eng. 2020;149(5):106854. doi:10.1016/j.cie.2020.106854.
152. Li X, Jiang P, Chen T, Luo X, Wen Q. A survey on the security of blockchain systems. Future Gener Comput Syst. 2020;107:841–53. doi:10.1016/j.future.2017.08.020.
153. Benaddi H, Ibrahimi K. A review: collaborative intrusion detection for IoT integrating the blockchain technologies. In: Proceedings of the 2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM); 2020 Oct 27–29; Reims, France. p. 1–6. doi:10.1109/wincom50532.2020.9272464.
154. Sakraoui S, Derdour M, Ahmim A. 6G-SECUREIDS: blockchain-enhanced secure knowledge transfer for distributed intrusion detection systems in advanced networks. In: Proceedings of the 2023 International Conference on Networking and Advanced Systems (ICNAS); 2023 Oct 21–23; Algiers, Algeria. p. 1–6. doi:10.1109/ICNAS59892.2023.10330507.
155. Liu H, Zhang S, Zhang P, Zhou X, Shao X, Pu G, et al. Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing. IEEE Trans Veh Technol. 2021;70(6):6073–84. doi:10.1109/TVT.2021.3076780.
156. Jin X, Ma C, Luo S, Zeng P, Wei Y. Distributed IIoT anomaly detection scheme based on blockchain and federated learning. J Commun Netw. 2024;26(2):252–62. doi:10.23919/JCN.2024.000016.
157. Xie N, Zhang C, Yuan Q, Kong J, Di X. IoV-BCFL: an intrusion detection method for IoV based on blockchain and federated learning. Ad Hoc Netw. 2024;163(4):103590. doi:10.1016/j.adhoc.2024.103590.
158. Alharthi H, Alshehri S, Kalkatawi M. Revolutionizing IoT security: a blockchain and federated learning-based anomaly detection system. In: Proceedings of the 2024 7th Artificial Intelligence and Cloud Computing Conference; 2024 Dec 14–16; Tokyo, Japan. p. 565–72. doi:10.1145/3719384.3719466.
159. Niu T, Liu Y, Li Q, Bao Q. An easily scalable docker-based privacy-preserving malicious traffic detection architecture for IoT environments. IEEE Access. 2024;12:191010–9. doi:10.1109/ACCESS.2024.3481496.
160. Kumar R, Kumar P, Tripathi R, Gupta GP, Garg S, Hassan MM. A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network. J Parallel Distrib Comput. 2022;164(2):55–68. doi:10.1016/j.jpdc.2022.01.030.
161. Akkal M, Cherbal S, Kharoubi K, Annane B, Gawanmeh A, Lakhlef H. An intrusion detection system for detecting DDoS attacks in blockchain-enabled IoMT networks. In: Proceedings of the 2024 7th International Conference on Signal Processing and Information Security (ICSPIS); 2024 Nov 12–14; Dubai, United Arab Emirates. p. 1–6. doi:10.1109/ICSPIS63676.2024.10812635.
162. Prasad VVSH, Bavirthi SS, Anupama CSS, Laxmi Lydia E, Kumar KS, Ammar K, et al. Blockchain enhanced distributed denial of service detection in IoT using deep learning and evolutionary computation. Sci Rep. 2025;15(1):22537. doi:10.1038/s41598-025-06568-8.
163. Alevizos L, Eiza MH, Ta VT, Shi Q, Read J. Blockchain-enabled intrusion detection and prevention system of APTs within zero trust architecture. IEEE Access. 2022;10(2):89270–88. doi:10.1109/ACCESS.2022.3200165.
164. Zhan X, Yuan H, Wang X. Research on block chain network intrusion detection system. In: Proceedings of the 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA); 2019 Sep 27–29; Xi'an, China. p. 191–6. doi:10.1109/iccnea.2019.00045.
165. Tug S, Meng W, Wang Y. CBSigIDS: towards collaborative blockchained signature-based intrusion detection. In: Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and

IEEE Smart Data (SmartData); 2018 Jul 30–Aug 3; Halifax, NS, Canada. p. 1228–35. doi:10.1109/Cybermatics_2018.2018.00217.

166. Li W, Tug S, Meng W, Wang Y. Designing collaborative blockchained signature-based intrusion detection in IoT environments. Future Gener Comput Syst. 2019;96(3):481–9. doi:10.1016/j.future.2019.02.064.
167. Ali Ujjan RM, Pervez Z, Dahal K. Snort based collaborative intrusion detection system using blockchain in SDN. In: Proceedings of the 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA); 2019 Aug 26–28; Island of Ulkulhas, Maldives. p. 1–8. doi:10.1109/SKIMA47702.2019.8982413.
168. Chelghoum M, Bendiab G, Labiod MA, Benmohammed M, Shiaeles S, Mellouk A. Blockchain and AI for collaborative intrusion detection in 6G-enabled IoT networks. In: Proceedings of the 2024 IEEE 25th International Conference on High Performance Switching and Routing (HPSR); 2024 Jul 22–24; Pisa, Italy. p. 179–84. doi:10.1109/HPSR62440.2024.10635989.
169. Stolz C, Zhang J. Blockchain-enabled distributed intrusion detection system for securing IoT networks. In: Proceedings of the MILCOM 2025–2025 IEEE Military Communications Conference (MILCOM); 2025 Oct 6–10; Los Angeles, CA, USA. p. 1–6. doi:10.1109/MILCOM64451.2025.11310020.
170. Xing R, Su Z, Wang Y. Collaborative intrusion detection approach based on blockchain in Internet of vehicles. IEEE Internet Things J. 2025;12(9):11965–76. doi:10.1109/JIOT.2024.3520615.
171. Pandya S. Robust blockchain and artificial intelligence-based efficient techniques for network intrusion detection system to improve cybersecurity. In: Proceedings of the 2025 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS); 2025 Feb 21–22; Raichur, India. p. 1–7. doi:10.1109/ICICACS65178.2025.10968395.
172. Chen Z, Huang J, Liu S, Long H. A blockchain and A-DCNN integrated framework for privacy protection and intrusion detection of industrial IoT. Computing. 2024;107(1):18. doi:10.1007/s00607-024-01390-y.
173. Nandanwar H, Katarya R. A hybrid blockchain-based framework for securing intrusion detection systems in Internet of Things. Clust Comput. 2025;28(7):471. doi:10.1007/s10586-025-05135-0.
174. Abou El Houda Z, Moudoud H, Brik B, Khoukhi L. Blockchain-enabled federated learning for enhanced collaborative intrusion detection in vehicular edge computing. IEEE Trans Intell Transp Syst. 2024;25(7):7661–72. doi:10.1109/TITS.2024.3351699.
175. Abdel-Basset M, Moustafa N, Hawash H, Razzak I, Sallam KM, Elkomy OM. Federated intrusion detection in blockchain-based smart transportation systems. IEEE Trans Intell Transp Syst. 2022;23(3):2523–37. doi:10.1109/TITS.2021.3119968.
176. Rahmadika S, Astillo PV, Choudhary G, Duguma DG, Sharma V, You I. Blockchain-based privacy preservation scheme for misbehavior detection in lightweight IoMT devices. IEEE J Biomed Health Inform. 2023;27(2):710–21. doi:10.1109/JBHI.2022.3187037.
177. Tukur YM, Thakker D, Awan IU. Edge-based blockchain enabled anomaly detection for insider attack prevention in Internet of Things. Trans Emerg Telecommun Technol. 2021;32(6):e4158. doi:10.1002/ett.4158.
178. Kumar P, Kumar R, Srivastava G, Gupta GP, Tripathi R, Gadekallu TR, et al. PPSF: a privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities. IEEE Trans Netw Sci Eng. 2021;8(3):2326–41. doi:10.1109/TNSE.2021.3089435.
179. Liu Y, Liu P, Jing W, Song HH. PD2S: a privacy-preserving differentiated data sharing scheme based on blockchain and federated learning. IEEE Internet Things J. 2023;10(24):21489–501. doi:10.1109/JIOT.2023.3295763.
180. Kumar C, Chittora P. Deep-learning and blockchain-empowered secure data sharing for smart grid infrastructure. Arab J Sci Eng. 2024;49(12):16155–68. doi:10.1007/s13369-024-08882-1.
181. Xiong H, Wu Y, Jin C, Kumari S. Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT. IEEE Internet Things J. 2020;7(12):11713–24. doi:10.1109/JIOT.2020.2999510.
182. Cui Q, Zhu Z, Ni W, Tao X, Zhang P. Edge-intelligence-empowered, unified authentication and trust evaluation for heterogeneous beyond 5G systems. IEEE Wirel Commun. 2021;28(2):78–85. doi:10.1109/MWC.001.2000325.
183. Athmani S, Bilami A, Boubiche DE. EDAK: an efficient dynamic authentication and key management mechanism for heterogeneous WSNs. Future Gener Comput Syst. 2019;92(2):789–99. doi:10.1016/j.future.2017.10.026.

184. Cao L, Liu Y, Cao S. An authentication protocol in LTE-WLAN heterogeneous converged network based on certificateless signcryption scheme with identity privacy protection. IEEE Access. 2019;7:139001–12. doi:10.1109/ACCESS.2019.2941913.
185. Panda SS, Jena D, Mohanta BK, Ramasubbareddy S, Daneshmand M, Gandomi AH. Authentication and key management in distributed IoT using blockchain technology. IEEE Internet Things J. 2021;8(16):12947–54. doi:10.1109/JIOT.2021.3063806.
186. Khalid U, Asim M, Baker T, Hung PCK, Tariq MA, Rafferty L. A decentralized lightweight blockchain-based authentication mechanism for IoT systems. Clust Comput. 2020;23(3):2067–87. doi:10.1007/s10586-020-03058-6.
187. Lin W, Zhang X, Cui Q, Zhang Z. Blockchain based unified authentication with zero-knowledge proof in heterogeneous MEC. In: Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops); 2021 Jun 14–23; Montreal, QC, Canada. p. 1–6. doi:10.1109/iccworkshops50388.2021.9473702.
188. García-Teodoro P, Camacho J, Maciá-Fernández G, Gómez-Hernández JA, López-Marín VJ. A novel zero-trust network access control scheme based on the security profile of devices and users. Comput Netw. 2022;212(1):109068. doi:10.1016/j.comnet.2022.109068.
189. Chen B, Qiao S, Zhao J, Liu D, Shi X, Lyu M, et al. A security awareness and protection system for 5G smart healthcare based on zero-trust architecture. IEEE Internet Things J. 2021;8(13):10248–63. doi:10.1109/JIOT.2020.3041042.
190. Cruz JP, Kaji Y, Yanai N. RBAC-SC: role-based access control using smart contract. IEEE Access. 2018;6:12240–51. doi:10.1109/ACCESS.2018.2812844.
191. Wang P, Xu N, Zhang H, Sun W, Benslimane A. Dynamic access control and trust management for blockchain-empowered IoT. IEEE Internet Things J. 2022;9(15):12997–3009. doi:10.1109/JIOT.2021.3125091.
192. Hu VC, Kuhn DR, Ferraiolo DF, Voas J. Attribute-based access control. Computer. 2015;48(2):85–8. doi:10.1109/MC.2015.33.
193. Han D, Zhu Y, Li D, Liang W, Souri A, Li KC. A blockchain-based auditable access control system for private data in service-centric IoT environments. IEEE Trans Ind Inform. 2022;18(5):3530–40. doi:10.1109/TII.2021.3114621.
194. Chen J, Tian Z, Cui X, Yin L, Wang X. Trust architecture and reputation evaluation for Internet of Things. J Ambient Intell Humaniz Comput. 2019;10(8):3099–107. doi:10.1007/s12652-018-0887-z.
195. Salamanis A, Kehagias DD, Tsoukalas D, Tzovaras D. Reputation assessment mechanism for carpooling applications based on clustering user travel preferences. Int J Transp Sci Technol. 2019;8(1):68–81. doi:10.1016/j.ijtst.2018.08.002.
196. Fu X, Yue K, Liu L, Feng Y, Liu L. Reputation measurement for online services based on dominance relationships. IEEE Trans Serv Comput. 2021;14(4):1054–67. doi:10.1109/TSC.2018.2854873.
197. Li Q, Malip A, Martin KM, Ng SL, Zhang J. A reputation-based announcement scheme for VANETs. IEEE Trans Veh Technol. 2012;61(9):4095–108. doi:10.1109/TVT.2012.2209903.
198. Shehada D, Gawanmeh A, Yeun CY, Zemerly MJ. Fog-based distributed trust and reputation management system for Internet of Things. J King Saud Univ Comput Inf Sci. 2022;34(10):8637–46. doi:10.1016/j.jksuci.2021.10.006.
199. Nwebonyi FN, Martins R, Correia ME. Reputation-based security system for edge computing. In: Proceedings of the 13th International Conference on Availability, Reliability and Security; 2018 Aug 27–30; Hamburg, Germany. p. 1–8. doi:10.1145/3230833.3232819.
200. Huang X, Yu R, Kang J, Zhang Y. Distributed reputation management for secure and efficient vehicular edge computing and networks. IEEE Access. 2017;5:25408–20. doi:10.1109/ACCESS.2017.2769878.
201. Guleng S, Wu C, Chen X, Wang X, Yoshinaga T, Ji Y. Decentralized trust evaluation in vehicular Internet of Things. IEEE Access. 2019;7:15980–8. doi:10.1109/ACCESS.2019.2893262.
202. Zhang H, Liu J, Zhao H, Wang P, Kato N. Blockchain-based trust management for Internet of vehicles. IEEE Trans Emerg Top Comput. 2021;9(3):1397–409. doi:10.1109/TETC.2020.3033532.
203. Li M, Tang H, Wang X. Mitigating routing misbehavior using blockchain-based distributed reputation management system for IoT networks. In: Proceedings of the 2019 IEEE International Conference on Communications Workshops (ICC Workshops); 2019 May 20–24; Shanghai, China. p. 1–6. doi:10.1109/iccw.2019.8757083.

204. She W, Liu Q, Tian Z, Chen JS, Wang B, Liu W. Blockchain trust model for malicious node detection in wireless sensor networks. IEEE Access. 2019;7:38947–56. doi:10.1109/ACCESS.2019.2902811.
205. Xiao L, Ding Y, Jiang D, Huang J, Wang D, Li J, et al. A reinforcement learning and blockchain-based trust mechanism for edge networks. IEEE Trans Commun. 2020;68(9):5460–70. doi:10.1109/tcomm.2020.2995371.
206. Yang Z, Wang R, Wu D, Yang B, Zhang P. Blockchain-enabled trust management model for the Internet of vehicles. IEEE Internet Things J. 2023;10(14):12044–54. doi:10.1109/JIOT.2021.3124073.
207. Ibrahim H, Ahakonye LAC, Lee JM, Kim DS. A unified AI-PureChain framework for verifiable intrusion prevention in industrial IoT systems. IEEE Internet Things J. 2026;13(7):12906–19. doi:10.1109/JIOT.2026.3652250.
208. Shit RC, Subudhi S. AI-powered anomaly detection with blockchain for real-time security and reliability in autonomous vehicles. In: Proceedings of the 2025 IEEE Space, Aerospace and Defence Conference (SPACE); 2025 Jul 21–23; Bangalore, India. p. 1–6. doi:10.1109/SPACE65882.2025.11170917.
209. Ahmed RH, Sultana J, Zahid S, Habib MA, Rauf A, Hussain M. Integrating large language models and AI into blockchain: a framework for intelligent smart contracts and fraud detection. IEEE Access. 2025;13(1):181323–35. doi:10.1109/ACCESS.2025.3622511.
210. Albshaier L, Budokhi A, Aljughaiman A. A review of security issues when integrating IoT with cloud computing and blockchain. IEEE Access. 2024;12(8):109560–95. doi:10.1109/ACCESS.2024.3435845.
211. Wan L, Eyers D, Zhang H. Evaluating the impact of network latency on the safety of blockchain transactions. In: Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain); 2019 Jul 14–17; Atlanta, GA, USA. p. 194–201. doi:10.1109/blockchain.2019.00033.
212. Chen X, Nguyen K, Sekiya H. On the latency performance in private blockchain networks. IEEE Internet Things J. 2022;9(19):19246–59. doi:10.1109/JIOT.2022.3165666.
213. Ferrag MA, Shu L. The performance evaluation of blockchain-based security and privacy systems for the Internet of Things: a tutorial. IEEE Internet Things J. 2021;8(24):17236–60. doi:10.1109/JIOT.2021.3078072.
214. Saini H, Bhushan B, Arora A, Kaur A. Security vulnerabilities in Information communication technology: blockchain to the rescue (a survey on Blockchain Technology). In: Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT); 2019 Jul 5–6; Kannur, India. p. 1680–4. doi:10.1109/ICICICT46008.2019.8993229.
215. Kushwaha SS, Joshi S, Singh D, Kaur M, Lee HN. Systematic review of security vulnerabilities in ethereum blockchain smart contract. IEEE Access. 2022;10(6):6605–21. doi:10.1109/ACCESS.2021.3140091.
216. Fu M, Wang P, Liu M, Zhang Z, Zhou X. IoV-BERT-IDS: hybrid network intrusion detection system in IoV using large language models. IEEE Trans Veh Technol. 2025;74(2):1909–21. doi:10.1109/TVT.2024.3402366.
217. He HY, Yang ZG, Chen XN. PERT: payload encoding representation from transformer for encrypted traffic classification. In: Proceedings of the 2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K); 2020 Dec 7–11; Ha Noi, Vietnam. p. 1–8.
218. Lin X, Xiong G, Gou G, Li Z, Shi J, Yu J. ET-BERT: a contextualized datagram representation with pre-training transformers for encrypted traffic classification. In: Proceedings of the ACM Web Conference 2022; 2022 Apr 25–29; Virtual Event. p. 633–42. doi:10.1145/3485447.3512217.
219. Liu Y, Wang X, Qu B, Zhao F. ATVITSC: a novel encrypted traffic classification method based on deep learning. IEEE Trans Inf Forensics Secur. 2024;19(2):9374–89. doi:10.1109/TIFS.2024.3433446.
220. Liu L, Li R, Li Q, Hou M, Jiang Y, Xu M. FlowletFormer: network behavioral semantic aware pre-training model for traffic classification. arXiv:2508.19924. 2025.
221. Chen XY, Han L, Zhan DC, Ye HJ. MIETT: multi-instance encrypted traffic transformer for encrypted traffic classification. Proc AAAI Conf Artif Intell. 2025;39(15):15922–9. doi:10.1609/aaai.v39i15.33748.
222. Sudasinghe PB, Liyanage KSK, Pussewalage HSG. Lightweight LLMs for network attack detection in IoT networks. In: Proceedings of the 2025 Computing, Communications and IoT Applications (ComComAp); 2025 Dec 14–17; Madrid, Spain. p. 395–400. doi:10.1109/comcomap68359.2025.11353200.
223. Algarni M, Dahab MY, Alsulami AA, Alturki B, Alghamdi BM, Alsemmeari RA, et al. An intrusion detection system based on lightweight BERT with meta-classifier for Internet of Things environments. Peerj Comput Sci. 2026;12(5):e3590. doi:10.7717/peerj-cs.3590.

224. Li Q, Zhang Y, Jia Z, Hu Y, Zhang L, Zhang J, et al. DoLLM: how large language models understanding network flow data to detect carpet bombing DDoS. arXiv:2405.07638. 2024.
225. Afifi F, Zaki F, Hanif H, Aqil N, Anuar NB. Transformer-based tokenization for IoT traffic classification across diverse network environments. PeerJ Comput Sci. 2025;11(1):e3126. doi:10.7717/peerj-cs.3126.
226. Masukawa R, Yun S, Jeong S, Huang W, Ni Y, Bryant I, et al. PACKETCLIP: multi-modal embedding of network traffic and language for cybersecurity reasoning. Front Artif Intell. 2025;8:1593944. doi:10.3389/frai.2025.1593944.
227. Peng W, Cui L, Cai W, Wang W, Cui X, Hao Z, et al. Bottom aggregating, top separating: an aggregator and separator network for encrypted traffic understanding. IEEE Trans Inf Forensics Secur. 2025;20(8):1794–806. doi:10.1109/TIFS.2025.3529316.
228. Shi Q, Shi D, He L, Xiao Q, Zheng L, Ma J, et al. TransGraphNet: robust detection of malicious encrypted network traffic *via* transformer and graph neural models. Peerj Comput Sci. 2025;11(6):e3353. doi:10.7717/peerj-cs.3353.
229. Qu J, Ma X, Li J. Trafficgpt: breaking the token barrier for efficient long traffic analysis and generation. arXiv:2403.05822. 2024.
230. Zhao D, Jiang B, Liu S, Cui S, Shen M, Han D, et al. Language of network: a generative pre-trained model for encrypted traffic comprehension. arXiv:2505.19482. 2025.
231. Sivaroopan N, Bandara D, Madarasingha C, Jourjon G, Jayasumana AP, Thilakarathna K. NetDiffus: network traffic generation by diffusion models through time-series imaging. Comput Netw. 2024;251(3):110616. doi:10.1016/j.comnet.2024.110616.
232. Gupta R, Liu S, Zhang R, Hu X, Wang X, Benkraouda H, et al. Generative active adaptation for drifting and imbalanced network intrusion detection. arXiv:2503.03022. 2025.
233. Wang T, Xie X, Zhang L, Wang C, Zhang L, Cui Y. ShieldGPT: an LLM-based framework for DDoS mitigation. In: Proceedings of the 8th Asia-Pacific Workshop on Networking; 2024 Aug 3–4; Sydney Australia. p. 108–14. doi:10.1145/3663408.3663424.
234. Li H, Kang H, Liu C, Wang R, Li J, Sun G, et al. Encrypted traffic detection in resource constrained IoT networks: a diffusion model and LLM integrated framework. IEEE Trans Netw Sci Eng. 2026;13(5):5324–44. doi:10.1109/TNSE.2025.3649259.
235. Huang A, Yan J, Fan X, Zhou H. Multi-scenario cloud-edge collaborative DDoS detection in LLM-enabled AIoT. IEEE Trans Netw Sci Eng. 2026;13:3790–809. doi:10.1109/TNSE.2025.3637740.
236. Park G, Lee S, Park Y. Minimizing response latency in LLM-based agent systems: a comprehensive survey. IEEE Access. 2026;14(4):26140–68. doi:10.1109/ACCESS.2026.3664226.
237. Huang L, Yu W, Ma W, Zhong W, Feng Z, Wang H, et al. A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ACM Trans Inf Syst. 2025;43(2):1–55. doi:10.1145/3703155.
238. Ben M, Kalganova T, Boulgouris N. A review of faithfulness metrics for hallucination assessment in large language models. IEEE J Sel Top Signal Process. 2025;19(7):1362–75. doi:10.1109/JSTSP.2025.3579203.
239. Kadhim EA, Feizi-Derakhshi MR, Aghdasi HS. NIASM: non-idiomatic abstractive summarization model to reduce hallucination and factual inconsistency problem in text generation. IEEE Access. 2026;14(6):67895–917. doi:10.1109/ACCESS.2026.3665264.