ENGINEERING Information Technology & Electronic Engineering
https://www.academax.com/EITEE; https://www.engineering.org.cn
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=11344065
E-mail: eng.itee@zju.edu.cn

Review

https://doi.org/10.1631/ENG.ITEE.2026.0095

Representation levels and objective consistency of network traffic generation: a survey

Biying WANG[1], Baosheng WANG[1], Shuang ZHAO[1,2], Jinshu SU[1,3], Shuhui CHEN[1], Minxin WANG[1], Zhengpeng LI[1], Ziling WEI[1]✉

[1]*College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China*
[2]*Department of Information Technology, Hunan Police Academy, Changsha 410138, China*
[3]*Academy of Military Sciences, Beijing 100091, China*

Abstract: Network traffic research relies on large-scale, high-quality traffic data. However, obtaining such data remains difficult because of privacy constraints, collection costs, class imbalance, and continuous updates. These challenges have increased researchers' interest in traffic generation. Although many generation methods have been proposed, existing studies and surveys often overlook two key questions: what form of traffic is generated and what practical objectives it can support. Based on 113 candidate records published from 2019 to 2026, this survey provides a detailed analysis of 39 representative network traffic generation studies through the lenses of representation levels and objective consistency. We organize existing methods into four representation levels and analyze how generated data relate to usage scenarios. We find that many methods preserve information that is useful for downstream tasks such as classification and intrusion detection, but task usefulness does not guarantee replayability or usability in real network environments. High-level representations are easier to model, yet they often discard protocol semantics, packet dependencies, and communication logic. We therefore distinguish task consistency from protocol consistency and show that the latter remains underexplored. We further summarize evaluation practices, discuss level-specific metrics, and highlight future directions including controllable generation, protocol-aware state-consistent synthesis, and engineering-oriented evaluation.

Key words: Network traffic generation; Traffic representation levels; Task consistency; Protocol consistency

1 Introduction

Research on network measurement and network intelligence increasingly depends on large-scale, high-quality traffic data. Many important tasks rely on real network traffic, including intrusion detection, malicious traffic identification, application classification, traffic prediction, and quality-of-service (QoS) analysis. However, acquiring such data has become increasingly difficult in practice (Chu et al., 2024; Wang BY et al., 2025).

First, as shown in Fig. 1, privacy protection and regulatory constraints limit access to many types of network data. In some cases, they also restrict the use of such data to narrowly defined purposes (Shahid et al., 2020; Kattadige et al., 2021). Releasing traffic data often requires additional anonymization effort, and data sharing is frequently restricted across organizations or even within departments. As a result, public access to network traffic data for research remains limited (Lin et al., 2020). Second, data annotation is costly. Many important tasks, such as application identification and malicious behavior detection, require fine-grained labels. However, such labels are often difficult to obtain. They usually require manual analysis or traffic replay (Sivaroopan et al., 2024). Third, many public datasets fail to reflect the evolving characteristics of network behavior. Models trained on these datasets often struggle to adapt to real-world network conditions. In

✉ Ziling WEI, weiziling@nudt.edu.cn
Biying WANG, https://orcid.org/0009-0005-0635-498X
Baosheng WANG, https://orcid.org/0009-0008-0583-6399
Shuang ZHAO, https://orcid.org/0000-0002-3423-8805
Jinshu SU, https://orcid.org/0000-0001-9273-616X
Shuhui CHEN, https://orcid.org/0000-0001-7413-8174
Minxin WANG, https://orcid.org/0009-0006-0743-9624
Zhengpeng LI, https://orcid.org/0009-0007-4146-9138
Ziling WEI, https://orcid.org/0000-0002-7858-1445
CLC number: TP393

Received: Apr. 5, 2026; Revision accepted: July 10, 2026;
Crosschecked: Aug. 3, 2026

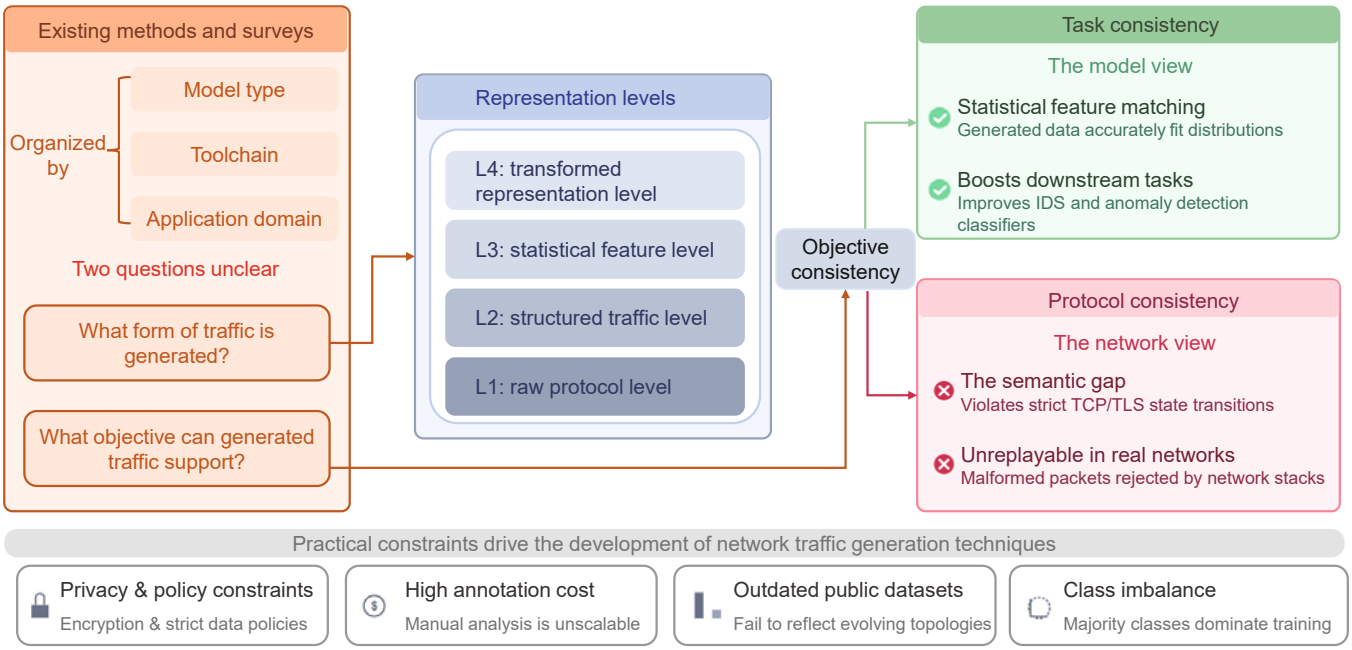

Fig. 1 Motivation and analytical framework of network traffic generation. IDS: intrusion detection system; TCP: transmission control protocol; TLS: transport layer security

addition, class imbalance is a common problem. In Internet of Things (IoT) scenarios, denial-of-service (DoS) attack traffic can greatly exceed other types of malicious traffic (Cai et al., 2025). As a result, majority classes may dominate the training process, while rare attacks are detected poorly. Severe class imbalance also exists in encrypted traffic datasets (Zhang HZ, 2024). This problem can significantly affect the performance of deep learning models.

Under these practical constraints, real data alone are no longer sufficient to support the development of network intelligence systems. As a result, network traffic generation has gradually evolved from a data augmentation technique into an important part of network research (Sivaroopan et al., 2024). As a data augmentation method, traffic generation can preserve key characteristics of the real traffic while introducing necessary variations. In this way, it can expand the scale and the diversity of datasets (Delgado-Soto et al., 2025). Traffic generation can simulate complex network environments and provide training data for machine learning models. For example, network security scenarios often require controllable synthetic malicious traffic that follows protocol specifications. Such traffic is useful for testing security tools and for developing robust detection methods (Huang et al., 2026). In addition, generative methods can generate diverse samples under new distributions if recent samples or updated protocol information is introduced as conditioning information. In this sense, traffic generation can help mitigate the limitation of outdated public datasets by expanding limited recent data and simulating evolving network scenarios.

Over the past decade, many studies have proposed network traffic generation methods based on generative adversarial networks (GANs) (Goodfellow et al., 2014), diffusion models (DMs) (Ho et al., 2020), and large language models (LLMs). These methods have been applied to tasks such as traffic classification and intrusion detection. Several surveys have reviewed this field from different perspectives. Zhang JH et al. (2015) introduced classical traffic generators and rule-based methods, while Adeleke et al. (2023) and Sivaroopan et al. (2026) discussed deep generative models and broader application scenarios. However, existing surveys mainly organize the literature by model type, toolchain, or application domain. They do not clearly distinguish what form of traffic is generated and what objective the generated traffic can support.

This distinction is important because different traffic representations preserve different types of information. A method that generates raw packets may retain packet fields and protocol structures, while a method that generates statistical features may only preserve information that is useful for a downstream model. In existing studies, generation quality is often evaluated by whether the generated data improve downstream model performance. This evaluation is useful for machine learning tasks, but it does not necessarily indicate that the generated traffic follows protocol rules, reflects communication behavior, or can be replayed in real network environments. Therefore, task consistency and protocol consistency should be treated as different objectives.

Motivated by this gap, this survey reviews network traffic generation from two related perspectives: representation levels and objective consistency. Representation levels describe what form of traffic is generated. For example, raw packets, packet or flow sequences, and statistical features retain different levels of protocol semantics, temporal dependencies, and replayability. Therefore, methods that are all described as traffic generation may in fact support different practical uses. Objective consistency further clarifies what the generated traffic is expected to support. In this survey, we distinguish task consistency from protocol consistency. Task consistency means that the

generated traffic preserves the patterns or information needed for downstream tasks, such as classification, detection, or prediction. Protocol consistency means that the generated traffic preserves protocol-valid structures, state transitions, and communication logic. These two forms of consistency are related, but they are not equivalent. A method may improve downstream performance while still failing to generate replayable or protocol-valid traffic.

This paper presents a systematic review of the literature on network traffic generation. The discussion is organized around the relationship between traffic representation choices and consistency objectives. Specifically, the main contributions of this paper are as follows:

1. We review representative studies on network traffic generation published from 2019 to 2026 and show that existing methods differ not only in model design, but also in the form of traffic they generate and in the practical objectives they support. On this basis, we propose an analytical framework based on traffic representation levels and objective consistency.

2. We organize network traffic generation into four representation levels and show a clear trade-off between information preservation and learnability. Lower-level representations remain closer to real protocol behavior and replayability, while higher-level representations are easier for generative models to learn but preserve less communication detail.

3. We distinguish task consistency from protocol consistency and show that many existing methods are mainly evaluated by downstream task improvement, while only a minority provide explicit evidence of protocol-consistent generation.

4. We summarize evaluation practices under different representation levels and explain how evaluation metrics should be interpreted under different generated representations. Based on this analysis, we suggest future directions in controllable generation, protocol-aware state-consistent synthesis, and engineering-oriented evaluation.

2 Methodology

This review follows a structured literature search and selection process to make the scope of the survey explicit and reproducible. The search focuses on studies whose primary topic is the generation, synthesis, augmentation, or replay-oriented construction of network traffic data. The search covers publications from 2019 to 2026.

We search major academic databases and publisher platforms, including IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, Wiley Online Library, Digital Bibliography & Library Project (DBLP), and Google Scholar. The search terms are designed to cover general and task-specific descriptions of network traffic generation. The main search queries include "network traffic generation," "synthetic network traffic," "synthetic network flow," "flow generation," "GAN network traffic," and "LLM network traffic generation." These terms are also combined with application-specific keywords such as intrusion detection, traffic classification, IoT traffic, and malicious traffic.

The inclusion criteria are as follows: first, the study must be a formally published journal article, conference paper, workshop paper, or book chapter within the selected time range; second, the generated or augmented object must be network traffic or a direct representation of network traffic, such as packets, flows, packet length sequences, statistical traffic features, traffic images, or traffic embeddings; third, the study must propose, evaluate, or systematically analyze a traffic generation, traffic augmentation, or replay-oriented traffic construction method; fourth, the study must provide enough technical detail to determine the representation level and the generation objective.

The exclusion criteria are as follows: we exclude studies that use the term "traffic" in a non-networking sense, studies focus only on traffic prediction without generating or augmenting traffic data, studies that only use generated data as an auxiliary tool without describing a traffic generation method, and studies whose generated objects are not network traffic or network-traffic-derived representations. Duplicate records are removed by comparing titles, authors, years, and digital object identifiers when available.

In the initial search, we identify 113 candidate records related to network traffic generation and closely related topics. These records are screened by checking titles, abstracts, keywords, and then full texts according to the inclusion and exclusion criteria. We focus on whether each study provides enough technical detail to determine its representation level, generation mechanism, and supported consistency objective. After this process, 39 representative studies are retained for detailed analysis and are summarized in Table 1.

3 Representation levels

The basic unit of network traffic is the packet. It contains protocol headers and payload data. Network traffic is usually identified by a five-tuple, including the source Internet protocol (IP) address, source port, destination IP address, destination port, and protocol. Packets with the same five-tuple are aggregated over time into a flow. For the transmission control protocol (TCP), a flow usually starts with the first SYN packet and ends with the last FIN or RST packet. For the user datagram protocol (UDP), it is often approximated as a group of packets with the same or reverse five-tuple within a fixed time window (Zhu et al., 2022).

Existing network traffic analysis methods are built on different data representations. Some methods directly use raw packets or raw bytes; some use packet or flow sequences; others use statistical features or transformed high-dimensional representations for tasks such as classification, detection, and prediction. These different representations are usually chosen based on task requirements. If the goal is traffic classification or anomaly detection, statistical features are sufficient. They are compact and easy for models to learn. However, if the goal is traffic replay, protocol testing, or realistic traffic simulation, the model must preserve protocol semantics, packet structure, and interaction states. In such cases, raw bytes are more suitable than features. In generation tasks, the representation determines what kind of traffic the model can actually generate and whether the generated data can be used in practice. A model that generates statistical features may help downstream learning, but it cannot directly reconstruct real packet exchanges. By contrast, a model that generates raw bytes can

Table 1 Summary of representative network traffic generation studies

Reference	Level	Model	Task consistency	Protocol consistency	Application domain
Charlier et al. (2019)	L3	GP-WGAN	✓	×	Intrusion detection
Ring et al. (2019)	L3	GAN	✓	×	Intrusion detection
Cheng (2019)	L1	CNN-based GAN	✓	✓	Intrusion detection
Dowoo et al. (2019)	L1	GAN	✓	✓	Intrusion detection augmentation
Lin et al. (2020)	L2	DoppelGANger	✓	×	Traffic prediction
Shahid et al. (2020)	L2	Autoencoder+WGAN	✓	×	IoT anomaly and intrusion detection
Manocchio et al. (2021)	L2	Manifold-guided GAN	✓	×	Intrusion detection
Meddahi et al. (2021)	L1	DCGAN+encoder/decoder	✓	✓	SIP security testing
Hui et al. (2022)	L2	Knowledge-enhanced GAN	✓	×	IoT traffic classification
Yin et al. (2022)	L2	DoppelGANger	✓	✓	Anomaly detection
Meslet-Millet et al. (2022)	L2	VAE, GMM, RNN	✓	✓	Traffic replay
Li J and Li (2022)	L3	EEMD-GAN	✓	×	Traffic prediction
Mozo et al. (2022)	L3	GAN	✓	×	Cryptomining attack detection
Nukavarapu et al. (2022)	L1	GAN-based packet generator	✓	×	DNS packet synthesis and security testing
Du et al. (2023)	L2	DBWE-Corbat	✓	×	Cyber-range training
Kholgh and Kostakos (2023)	L1	LLM-based (GPT-3)	✓	✓	Intrusion detection
Qi et al. (2024)	L2	Conditional DM	✓	×	5G network planning
Zhang HZ (2024)	L2	Transformer and relative GAN	✓	×	Encrypted traffic classification
Jiang et al. (2024)	L4	Stable DM	✓	✓	Traffic classification augmentation
Li T et al. (2024)	L2	Multi-scale hierarchical GAN	✓	×	Network optimization
Chu et al. (2024)	L1	State space model	✓	✓	Traffic classification
Soper et al. (2024)	L1	Two-stage payload simplification	✓	×	Intrusion detection
Chai et al. (2024)	L2	Denoising diffusion probabilistic model	✓	×	Network optimization
Poisson et al. (2024)	L1	Node behavior simulation in a real topology	✓	✓	IoT anomaly and intrusion detection
Zhang SY et al. (2024)	L3	Service-guided hierarchical DM	✓	×	Network optimization
Sivaroopan et al. (2024)	L4	DMs	✓	×	Traffic fingerprinting and anomaly detection
Wang MX et al. (2024)	L2	GAN	✓	×	Intrusion detection
Chai et al. (2025a)	L2	Spatiotemporal knowledge-driven DM	✓	×	Network optimization
Chai et al. (2025b)	L2	Transformer-based DM	✓	×	Wireless network optimization
Shin et al. (2025)	L1	Conditional GAN	✓	×	Encrypted traffic classification
Oh et al. (2025)	L2	Temporal shifting, spatial correlation sampling, and interpolation	✓	×	Encrypted traffic classification
Sun PS et al. (2025)	L1	Meta-Llama-3-8B (Meta AI, 2024)	✓	✓	Malicious traffic detection
Zhang S et al. (2025)	L1	MUD-based behavior grammars	✓	✓	IoT security testing
Cai et al. (2025)	L4	TransGAN and spatial-channel synergistic self-attention	✓	×	Malicious traffic detection
Delgado-Soto et al. (2025)	L1	GPT-3.5	✓	✓	Security evaluation
Chu et al. (2026)	L1	State space model	✓	✓	Service recognition and anomaly detection
Jonath et al. (2026)	L3	GMM and conditional GAN	✓	×	Encrypted malware classification
Huang et al. (2026)	L1	Commercial LLM	✓	✓	Intrusion detection
Jablaoui and Liouane (2026)	L3	Genetic algorithm, 1D-CNN, and BiGRU	✓	×	IDS augmentation

GP-WGAN: Wasserstein GAN with gradient penalty; CNN: convolutional neural network; DCGAN: deep convolutional GAN; VAE: variational autoencoder; GMM: Gaussian mixture model; RNN: recurrent neural network; SIP: session initiation protocol; DNS: domain name system; BiGRU: bidirectional gated recurrent unit; IDS: intrusion detection system; MUD: manufacturer usage description; GPT: generative pre-trained Transformer

better support traffic replay and protocol-aware applications. Therefore, the representation level is not only a description of data form, but also a key basis for understanding the capability and target of a generation method.

From this perspective, we organize network traffic representations into four levels, as shown in Fig. 2.

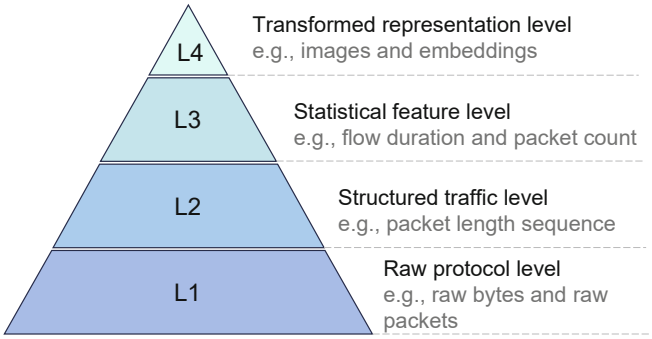

Fig. 2 Network traffic representation levels

Raw protocol level (L1) refers to representations based on raw bytes and raw packets that preserve protocol semantics.

Structured traffic level (L2) refers to packet-level or flow-level sequence representations that retain communication timing and conversation structure, such as packet length sequences and inter-arrival time sequences.

Statistical feature level (L3) refers to statistical representations obtained through feature extraction, such as flow duration and packet count.

Transformed representation level (L4) refers to high-dimensional representations derived from the features such as images or embeddings.

To make the level assignments in Table 1 reproducible, each method is classified according to the representation that is directly generated by a generation model, rather than according to the source data used for preprocessing or the final downstream task. If the model directly generates packet bytes, protocol fields, or executable packets, it is classified as L1. If it generates ordered packet or flow sequences, such as packet lengths, inter-arrival times, directions, or flow events, it is classified as L2. If it generates aggregated statistical feature vectors, it is classified as L3. If it generates transformed spaces such as images, it is classified as L4.

3.1 Raw protocol level

Raw protocol level represents the original protocol bytes of network traffic. It contains the packet bytes captured from network interfaces. Byte data often exhibit clear structural characteristics due to the fixed relationship between protocol headers and payloads. This structure provides a clear mapping between byte data and protocol fields (Shin et al., 2025). Raw traffic is usually captured in packet capture (PCAP) format, which preserves complete packet transmission records. However, PCAP files are large and cannot be used as direct input to neural networks. They usually need to be converted into standardized representations through preprocessing (Shapira and Shavitt, 2021).

Network traffic generation at L1 usually follows protocol specifications to ensure structural integrity and protocol consistency. Delgado-Soto et al. (2025) used packet definitions extracted from Wireshark to construct the training data. Through fine-tuning and prompt engineering, the proposed system successfully generates packet-level and conversation-level network traffic for Internet control message protocol (ICMP), address resolution protocol (ARP), domain name system (DNS), TCP, and hypertext transfer protocol (HTTP). This method can simulate real-world network conversations without packet or protocol errors. TrafficT5 (Huang et al., 2026) achieves high-fidelity generation through a three-stage self-correcting process. It first predicts flow-level features, and then generates byte-aligned hex under a fixed 00–FF vocabulary. Finally, TrafficT5 applies a repair module to enforce protocol invariants and performs detector-guided, iterative byte-level correction trained with multi-task objectives.

3.2 Structured traffic level

Structured traffic level refers to traffic representations that preserve packet-level or flow-level temporal structure. It encodes packet or flow sequences in a structured form while maintaining the original time order and inherent relationships of the data. This representation allows the dynamic process of network communication to be directly reflected. It also provides effective inputs for traffic analysis and generation tasks.

Lin et al. (2020) proposed the DoppelGANger framework, which uses long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) to generate batches of temporal subsequences. This design reduces the number of single-step transmissions and improves the ability to capture long-term temporal dependencies. It also addresses the limitation of traditional step-by-step recurrent neural network (RNN) generation, where dependencies may be lost when modeling long sequences. Wang MX et al. (2024) adopted a hierarchical structure that combines flow-level five-tuple information with packet-level feature sequences. It avoids the dimensional conflicts caused by traditional feature vectors and ensures the functional validity of communication behavior and malicious traffic characteristics.

3.3 Statistical feature level

Statistical feature level refers to representations built from statistical measures and behavioral summaries extracted from packet-level or flow-level traffic data. Common examples include packet length, packet rate, inter-arrival time, flags, and header- or payload-related features. This form of representation converts raw traffic into structured feature vectors. Compared with raw traffic, it provides a more compact description and is easier to use in analysis and modeling.

Shafi et al. (2025) selected key statistical features and mined feature combination patterns. They represented different network activities as distinguishable behavior profiles or pattern dictionaries. This method allows activity recognition and classification in a unified feature space. In this type of method, statistical features are used not only for classifier training, but also as statistical evidence to describe behavioral differences. For example, density estimation can be used

to characterize the distribution range of feature values under different activities (Chen, 2017). These studies further use association rule mining to analyze relationships between features. Optimization strategies are also applied to extract joint patterns among features (Kennedy and Eberhart, 1995; Shawkat et al., 2022).

Ring et al. (2019) studied mixed attributes in network traffic. These attributes include continuous and categorical variables. They proposed three preprocessing methods to convert these attributes into continuous values. The first method is direct numerical encoding and normalization. The second method expands the attributes into binary bit vectors. The third method uses embedding-based representations. For example, an IP address can be mapped to a continuous vector. In this way, semantically similar IP addresses are placed close in the vector space. These methods help preserve the statistical structure of network behavior and make the data suitable for generative modeling. In addition, the statistical feature space is often combined with distribution-based measures. It can be used together with interpretable analysis methods. Researchers can use distribution distances to measure heterogeneity across different data sources or clients (Carillo et al., 2025).

3.4 Transformed representation level

Transformed representation level maps network traffic into more abstract representation spaces than those used at L2 and L3. These methods do not model the raw bytes or packet sequences directly. Instead, they first apply transformations to obtain new representations, such as images. This helps capture temporal correlations, long-range dependencies, and high-dimensional relationships. These representations are suitable for learning and modeling.

Converting traffic data into image data is a common data processing method in network traffic research. NetDiffus (Sivaroopan et al., 2024) converts one-dimensional (1D) traffic sequences into two-dimensional (2D) Gramian angular summation field (GASF) images and then performs generation in the image domain using DMs. The generated representation is an image. Therefore, NetDiffus is categorized as an L4 method. Related studies also show that images can reveal structural properties such as multi-scale bursts and long-range dependence, while providing a visually organized basis for abstract traffic analysis (Fernandes et al., 2015).

In addition, by using flow spectrum, Yang et al. (2025) mapped the raw network flow from a high-dimensional space to a 1D spectral space. There are abstract representations centered on distributions and marginal statistics. For example, high-dimensional traffic attributes can be compressed by constructing marginal distributions. Sun DY et al. (2024) preserved key correlations while introducing noise to satisfy differential privacy. At the same time, features such as time intervals can be incorporated into binning and marginal selection so that temporal information is retained.

3.5 Analysis of representation levels

As shown in Table 1, we summarize representative studies on network traffic generation in recent years and compare them across different representation levels. Task consistency is marked when the original study reports downstream task evidence for the generated traffic, such as classification, detection, prediction, optimization, or security evaluation results. Since most reviewed studies are designed for task-oriented traffic generation, this column is widely marked. Its purpose is to indicate reported task-level utility rather than to rank methods by downstream performance. Protocol consistency is marked when the original study reports protocol-level evidence, such as packet parsing, replay success, or packet success rate. For example, generated flow features may improve intrusion detection accuracy and therefore provide task consistency. However, if these features cannot be converted into valid packets or replayed in a real network stack, they do not provide protocol consistency.

The representation level is the conceptual basis of this survey because it determines what a generation method actually generates. A method that generates packet bytes and a method that generates traffic images may both be described as traffic generation methods. However, their outputs preserve different information and support different practical uses. Therefore, representation level is not only a data-format issue, but also a key factor that determines whether the generated traffic can support downstream learning, protocol validation, traffic replay, or realistic network simulation.

In this study, we compare traffic representation levels from four aspects: information preservation, compliance, replayability, and learnability. This comparison is a qualitative analysis derived from the type of information retained by each representation level, rather than an empirical score or an absolute ranking. The four comparison dimensions are defined as follows:

1. Information preservation describes how much information a representation retains from the original communication process. It reflects the degree of irreversible information loss introduced when traffic is transformed from bytes to high-level representations.

2. Compliance describes whether a representation still follows the communication rules of network protocols, such as TCP and transport layer security (TLS). It also reflects whether the representation can still correspond to a valid communication process.

3. Replayability refers to whether the representation can be mapped back to the real network traffic and can be accepted and executed by an actual network stack.

4. Learnability describes how well the representation fits the modeling capacity of generative models.

L1 methods, such as PAC-GAN (Cheng, 2019), GPT on the wire (Delgado-Soto et al., 2025), and TrafficT5 (Huang et al., 2026), operate close to packet bytes or protocol fields. They preserve enough structural information to support packet parsing, protocol checks, and, in some cases, executable replay. However, they must satisfy strict syntax constraints and byte-level dependencies during generation. L2 methods, such as DoppelGANger (Lin et al., 2020), NetShare (Yin et al., 2022), and ProGen (Wang MX et al., 2024), preserve ordered communication structures, timing patterns, or flow evolution. These methods are useful for workload modeling and flow-level behavior analysis. However, they usually require

additional reconstruction or constraints before they can become replayable packet traces. L3 methods, such as SynGAN (Charlier et al., 2019) and ensemble empirical mode decomposition (EEMD)-GAN (Li J and Li, 2022), operate in compact feature spaces and are therefore easier to optimize. EEMD decomposes complex traffic sequences into simpler temporal components. However, these methods mainly preserve prediction- or classification-oriented information rather than executable communication logic. L4 methods, such as NetDiffus (Sivaroopan et al., 2024) and classifier TransGAN and spatial-channel synergistic self-attention (CT-SSSA) (Cai et al., 2025), are even farther from the original traffic process, because the model learns image-like or transformed spaces. This makes them convenient for learning and augmentation, but their outputs usually cannot be directly interpreted as replayable traffic. Based on the introduction of the representation levels, we summarize the characteristics of each level in Table 2. As the representation level moves from L1 to L4, network traffic becomes more abstract. During this process, packet-level details, protocol structures, temporal relationships, and interaction behaviors are gradually compressed or removed. As a result, information preservation, compliance, and replayability generally become weaker. In contrast, learnability usually improves because higher-level representations are more compact, regularized, and easier to fit with modern generative models.

The loss of information can be illustrated with a simple TCP three-way handshake. At L1, the representation can retain the actual packet bytes and field values of the SYN, SYN-ACK, and ACK packets, including flags, sequence numbers, header lengths, and payload presence. At this level, a generator can generate traffic that is checked by protocol parsers or replayed by a network stack. At L2, the same handshake may be represented only as an ordered sequence of packet lengths, directions, timestamps, or flow events. This preserves the fact that the interaction contains three ordered steps, but it may no longer preserve the exact TCP option values or byte-level dependencies between sequence and acknowledgment numbers. At L3, the handshake is reduced further to summary statistics, such as packet count, mean packet size, duration, forward-to-backward packet ratio, or total bytes. The existence of a handshake can only be inferred indirectly through aggregated behavior. At L4, the same event may appear only as an image. Such a representation may help distinguish benign and malicious communication patterns, but the protocol semantics needed for direct interpretation or replay have disappeared.

This three-way handshake also clarifies the relationship between compliance and protocol consistency. A representation may remain compliant in the sense that it preserves a meaningful structure for that level, such as a valid packet length sequence or a coherent traffic image. However, protocol consistency requires more than structural plausibility. It requires evidence that the generated result corresponds to executable communication behavior, such as correct packet parsing, successful replay, low bad-packet rate (BPR), or state-consistent interaction across packets. For this reason, compliance is used in Table 2 as a property of the representation, whereas protocol consistency in Table 1 is used as an objective supported by explicit evidence.

4 Network traffic generation models

Section 3 focuses on what form of traffic is generated, whereas this section focuses on how traffic is generated. The same model may operate on different representation levels and support different consistency objectives. For this reason, this review does not focus only on model architectures. It also examines the representation levels on which these models operate and the types of consistency they can realistically support. Existing network traffic generation methods can be grouped into four categories based on their generation mechanisms: rule-based and data transformation methods, GAN-based methods, DM-based methods, and LLM-based methods. These methods learn traffic distribution patterns from different perspectives and can operate on different representation levels. This section reviews these methods and analyzes them alongside their corresponding representation levels and consistency objectives.

4.1 Rule-based and data transformation methods

Rule-based and data transformation methods are early approaches to network traffic generation. Instead of learning a full generative distribution from data, these methods create samples by modifying existing traffic records through rules or structure-preserving transformations. They are often used for traffic augmentation, especially to increase sample diversity, reduce class imbalance, and simulate variations of known traffic patterns. A common scenario is intrusion detection, where minority classes are often underrepresented.

Table 2 Qualitative comparison of network traffic representation levels

Level	Information preservation	Compliance	Replayability	Learnability
L1	Packet-level information	Explicit protocol structure	Directly replayable if valid	Difficult (bytes are high-dimensional and syntax-constrained)
L2	Sequential traffic structure	Partial communication structure	Requires packet reconstruction	Moderate (temporal dependencies must be learned)
L3	Statistical traffic features	Mostly abstracted	Not directly replayable	Relatively easy (compact feature vectors)
L4	Transformed or embedded features	Indirect or absent	Not directly replayable	Relatively easy (regularized representation spaces)

Traditional oversampling methods such as synthetic minority over-sampling technique (SMOTE) (Chawla et al., 2002) are widely used, but they may distort feature relationships and fail to preserve traffic semantics. To address this issue, Jablaoui and Liouane (2026) proposed a data augmentation method based on a genetic algorithm. The method generates minority samples through one-point crossover and Gaussian noise mutation, producing traffic that is similar in structure and diverse. It uses cosine similarity as the fitness function to ensure that the generated samples remain semantically consistent with the original data. Compared with traditional interpolation methods, this approach produces samples that are closer to the real data distribution while offering better interpretability and lower computational cost.

As research progressed, data augmentation techniques specifically designed for network traffic have been proposed. Script-driven traffic generators create packets from predefined traces, rules, or configuration patterns, allowing users to modify packet header fields, payload content, packet size, and timing logic (Adeleke et al., 2023). Other methods simulate network variation through structured modification. For example, average augmentation constructs samples from aligned temporal observations within the same class, while maximum transmission unit (MTU) augmentation simulates different MTU settings through packet splitting and reassembly (Zion et al., 2025). These methods do not model the full communication process, but they can generate controlled variations of existing traffic. Dowoo et al. (2019) proposed another method that transforms time differences between adjacent packets into interval images and applies sequence-aware preprocessing to reduce repeated patterns. Although this method is combined with a GAN framework, its preprocessing stage follows the logic of traffic transformation. It restructures existing traffic signals rather than synthesizing protocol-valid communication.

From the perspective of representation levels, rule-based and data transformation methods mainly operate at L2 and L3. At L3, they modify feature vectors through interpolation, recombination, or feature-space mutation. At L2, they operate on packet- or flow-level sequences by adjusting packet timing, packet size patterns, or temporal order while preserving coarse communication structures. These methods are mainly designed to improve downstream tasks such as classification, detection, and prediction. Therefore, they primarily support task consistency rather than protocol consistency. In addition, rule-based and data transformation methods do not require end-to-end training of large generative models. They are relatively easy to deploy and are often suitable for quick dataset augmentation under limited computational resources.

4.2 GAN-based methods

GANs have been widely applied in network traffic data augmentation because of their strong feature learning ability and advantages in modeling high-dimensional data distributions (Goodfellow et al., 2014). As shown in Fig. 3, a GAN consists of two components: a generator and a discriminator. The generator takes random noise as input and aims to produce synthetic samples that are similar to the real data. The discriminator distinguishes between real and generated samples. Through adversarial training, the two components are optimized iteratively. This process enables the model to learn to generate data that follow the distribution of real samples without strong prior assumptions about the data structure. This makes GANs suitable for modeling network traffic data with high feature dimensions, complex structures, and diverse sample types. However, traditional GANs still suffer from problems such as mode collapse, training instability, and gradient vanishing (Sivaroopan et al., 2024).

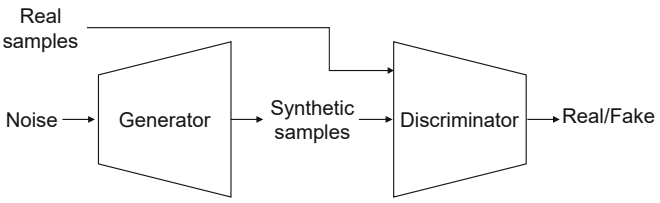

Fig. 3 Structure of a GAN

Early GAN-based traffic generation studies mainly focused on feature-level or sequence-level augmentation. In these settings, the goal is usually to improve downstream learning performance. For example, Charlier et al. (2019) introduced the Wasserstein GAN with gradient penalty (GP-WGAN) to generate distributed DoS (DDoS) attack traffic. The generated attack data are highly similar to real data in key features such as mean packet length and traffic bytes per second. In addition, GP-WGAN can effectively reduce mode collapse and improve the diversity of generated samples. Manocchio et al. (2021) proposed a manifold-guided GAN framework to improve diversity in synthetic flow generation for intrusion detection. Zhang HZ (2024) further combined Transformer-based self-attention with a relative discriminator to better model long-range dependencies in encrypted traffic sequences. These methods illustrate the main role of GANs in task-oriented traffic generation. They are effective at learning complex distributions at L2 or L3 and are often used to enhance classification, detection, or prediction performance. In addition, GAN-based methods usually have fast sampling and can generate synthetic traffic in a single forward pass, which makes them attractive for large-scale augmentation.

Some GAN-based studies move toward lower-level traffic generation by preserving the communication structure. For example, Lin et al. (2020) used DoppelGANger to generate structured temporal subsequences and improve the modeling of long-range traffic dependencies. Wang MX et al. (2024) combined flow-level five-tuple information with packet-level feature sequences in a hierarchical framework, which helps retain more communication-related structure than purely statistical feature generation. Other work goes further toward raw packet generation. Cheng (2019) encoded packet bytes into a matrix and applied a convolutional neural network (CNN)-based GAN to generate IP packets, reporting high transmission success rates for several packet types such as ICMP, DNS, and HTTP. Similarly, Meddahi et al. (2021) focused on session initiation protocol (SIP) traffic generation under protocol-aware constraints. These studies show that GANs can be extended to the generation process at L1, but doing so usually requires additional structural encoding, protocol repair, or explicit validity

constraints beyond standard adversarial training.

In addition to GAN-based approaches, latent-variable generative models based on variational autoencoders (VAEs) (Kingma and Welling, 2014) have been explored in network traffic generation. NeCSTGen (Meslet-Millet et al., 2022) combines a VAE, a Gaussian mixture model (GMM), and recurrent generation to synthesize realistic network traffic and replayable PCAP traces. Compared with adversarial models, VAE-based methods are usually easier to train and provide a smoother latent space for structured traffic generation. In this method, the generation process operates at L2, where the generated object is a structured traffic sequence or trace.

From the perspective of the representation-level framework, GAN-based methods are most commonly used at L2 and L3, where they generate packet or flow sequences, traffic features, or intermediate structured representations. In these settings, they mainly support task consistency, because the generated outputs are designed to preserve the information needed for detection or classification rather than communication behavior. A smaller number of GAN-based methods operate at L1 by generating packet-level or byte-level representations. These approaches are closer to protocol consistency. However, they show a key limitation of GANs in traffic generation. Adversarial learning alone cannot ensure valid protocol syntax, cross-field consistency, state transitions, or replayability. Therefore, GAN-based methods are useful mainly for task-oriented distribution learning. Their main limitation is that they still need constraints or structured decoding to produce executable traffic.

4.3 DM-based methods

DMs (Ho et al., 2020) have become one of the major techniques in network traffic generation because of their strong training stability, high sample quality, and flexible controllability. They are generative models based on a Markov chain and consist of a forward noising process and a reverse denoising process. As shown in Fig. 4, this illustration presents the process of forward noising and reverse denoising for traffic grayscale images. The forward process gradually adds Gaussian noise to a real sample over multiple time steps, so that the sample progressively loses its original structure and eventually approaches an isotropic standard normal distribution. This process is fixed and does not require learning. The reverse process is learned by a parameterized denoising network, which starts from pure noise and progressively removes noise step by step to recover samples following the real data distribution. This progressive generation mechanism usually provides stable training and high sample quality. DM variants further improve the fidelity and downstream usefulness of generated traffic by introducing conditional information and specialized network modules.

Existing DM-based traffic generation studies mainly operate on structured or transformed representations rather than directly on raw protocol bytes. At L2, DMs are used to model network evolution, flow variation, or traffic demand patterns. For example, Qi et al. (2024) proposed a conditional diffusion framework for 5^{th} generation mobile communication technology (5G) traffic generation, where regional 4^{th} generation mobile communication technology (4G) traffic features are used as conditional information to guide the generation of 5G spatiotemporal traffic patterns. Chai et al. (2024) developed a diffusion-based framework for mobile traffic generation using public geographic and demographic auxiliary information, showing that DMs can support controllable generation when traffic is represented at an aggregated spatiotemporal level. Chai et al. (2025a) further extended this line of work through spatiotemporal knowledge-driven diffusion modeling.

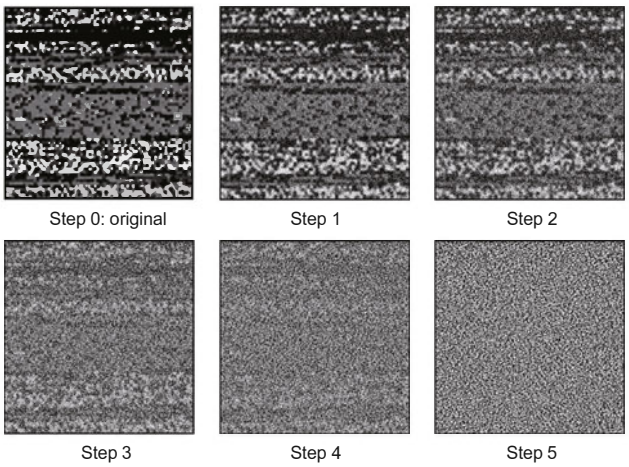

Fig. 4 Illustration of the forward noising and reverse denoising processes in DMs

DMs are widely used in transformed representation spaces, especially when traffic is mapped into an image. NetDiffus (Sivaroopan et al., 2024) converts 1D traffic sequences into 2D GASF images. GASF transforms sequential traffic data into an image-like representation, after which generation is performed in the image domain using DMs. This design allows DMs to operate on a more regular representation space, which can improve learnability and support effective task-oriented generation for traffic analysis. The value of DMs lies less in protocol-level realism than in their ability to learn abstract structures and class-related distributional patterns.

A few diffusion-based methods move closer to protocol-aware generation. NetDiffusion (Jiang et al., 2024) applies DMs to packet-related traffic representations and uses controlled generation to preserve protocol field distributions. This method shows that DMs are not limited to high-level transformed representations. However, when DMs are used for lower-level packet or protocol generation, they face challenges similar to GANs. The model must preserve field constraints, packet dependencies, and communication logic, not only the overall data distribution. In practice, this often requires protocol-specific repair or constrained decoding.

Compared with GAN-based methods, as shown in Table 3, DMs are used when the task needs stable generation, diverse samples, or better distribution fitting, rather than fast generation. The denoising process of DM-based methods can avoid some instability caused by adversarial training. Their probabilistic design makes them suitable for complex traffic patterns.

From the perspective of the representation-level framework, DM-based methods are mostly used at L2 and L4, while

Table 3 Comparison between GANs and DMs in network traffic generation

Property	GANs	DMs
Principle	Adversarial training	Noising–denoising generation
Training stability	Sensitive to training balance and mode collapse	Generally stable optimization
Generation process	Single-pass sampling	Iterative sampling
Sampling speed	Fast	Slow
Typical level	Mainly at L2 or L3; some at L1	Mainly at L2 or L4; some at L1 or L3
Protocol consistency	Limited without extra constraints	Limited without semantic control
Generation diversity	May be reduced by mode collapse	Generally strong
Training data requirement	Moderate, but tuning-sensitive	Often higher and more structured
Replayability support	Weak without protocol validation	Weak without repair or control

they are used at L3 and L1 in some studies. At L2 and L3, they often generate temporal traffic patterns or aggregated demand patterns for planning, optimization, and prediction. At L4, they work well when traffic is converted into images or abstract representations. This improves learnability but reduces replayability. Therefore, most DM-based methods support task consistency rather than protocol consistency. Their strength is stable distribution modeling across complex representations. Their main limitation is that they need semantic constraints, repair mechanisms, or explicit modeling of communication structure to generate replayable, state-consistent, and protocol-valid traffic.

4.4 LLM-based methods

LLMs provide another way to generate network traffic. They can generate traffic based on prompts, protocol descriptions, structured fields, or user intents. This makes them useful when users need controllable traffic generation, such as generating traffic for attack simulation, protocol testing, or security evaluation.

One advantage of LLM-based methods is that they can use structured prior knowledge during generation. They can be guided by protocol specifications, field templates, or attack descriptions. For example, GPT on the wire (Delgado-Soto et al., 2025) uses packet definitions extracted from Wireshark. It combines fine-tuning and prompt engineering to generate packet-level and conversation-level traffic for protocols such as ICMP, ARP, DNS, TCP, and HTTP. PAC-GPT (Kholgh and Kostakos, 2023) explores the generative pre-trained Transformer (GPT) language modeling for synthetic traffic generation. These studies show that LLMs can use textual and symbolic guidance, rather than only learning statistical patterns from data.

Recent studies have further explored executable traffic generation. TrafficT5 (Huang et al., 2026) proposes a multi-stage self-correcting framework that converts natural language intents into executable PCAP files. It first predicts flow-level features, then generates byte-aligned hexadecimal sequences, and finally applies protocol-aware repair and consistency correction. In this setting, BPR is the fraction of packets that fail round-trip parsing or validation (Huang et al., 2026). TrafficT5 reports a low BPR, showing that its repair and correction steps improve packet-level validity. This type of framework is closely related to protocol consistency, because it checks not only statistical similarity or task performance, but also whether the generated traffic can be executed in a real network environment. LLM-based methods are also used for adversarial generation. For example, adversarial traffic generation (AdvTG) (Sun PS et al., 2025) combines an LLM with reinforcement learning to generate adversarial malicious traffic that can evade detection models. Instead of modeling only the traffic distribution, it conditions generation on functional fields and target types. It then modifies non-functional fields to improve adversarial effectiveness. This shows that LLMs can support generation under semantic control, which is difficult for traditional feature space generators.

From the perspective of the representation-level framework, LLM-based methods are concentrated mainly at L1. Many of them aim to generate packet bytes, protocol fields, packet sequences, or executable artifacts such as PCAP files. This gives them stronger potential to support protocol consistency than many GAN-based or DM-based methods at L2–L4. However, this potential does not come from the language model alone. It depends on structural guidance and protocol-aware constraints. Therefore, the strength of LLM-based traffic generation lies in controllable and intent-aware synthesis. Its main limitation is the mismatch between language model tokenization and network protocol structure. Without constraints, repair, and validation mechanisms, LLMs may generate outputs that are useful for task-oriented analysis but not reliable for protocol-consistent replay. Methods based on LLMs usually incur high practical costs because they often require substantial computing resources and large-scale training or inference resources.

At the same time, LLM-based traffic generation faces several technical challenges that are more fundamental than those encountered in natural language generation. First, tokenization is not naturally aligned with packet bytes or protocol fields. A standard text tokenizer does not preserve fixed byte boundaries, protocol header structures, or field-level dependencies. Second, byte-level alignment is difficult. Even if a model generates hexadecimal strings, correct field lengths, checksums, sequence numbers, acknowledgment numbers, and cross-field constraints must still be preserved. Third, protocol validity is inherently stateful. Multi-packet communication depends not only on local packet correctness, but also on timing, ordering, state transitions, and interaction logic across sessions. As a result, unconstrained token-by-token decoding can easily

produce malformed packets or structurally inconsistent conversations. These challenges explain why recent LLM-based methods often rely on structured intermediate representations, repair modules, tool feedback, or iterative self-correction rather than direct free-form generation alone.

5 Evaluation metrics

Evaluation of network traffic generation is closely related to representation levels and generation objectives. A method that generates aggregated traffic features should not be assessed in the same way as a method that aims to produce replayable packet traces. Therefore, this section summarizes the evaluation metrics used in the literature and explains how their applicability varies across different representation levels.

The choice of evaluation metrics depends on the representation level and the generation method. Different representation levels preserve different kinds of traffic information, so they cannot be assessed in the same way. In existing studies, evaluation metrics for network traffic generation can generally be grouped into two dimensions: similarity and usability. Similarity describes how close the generated data are to the real data in terms of statistical distributions or numerical error. Usability describes whether the generated data are beneficial for downstream tasks or valid at the protocol level.

Let T denote the total number of evaluated samples, t the sample index, y the real sample set, and $\hat{y}$ the generated sample set. These symbols are used consistently in the following metric definitions.

5.1 Similarity metrics

Similarity metrics generally follow three evaluation paths: distribution consistency, point-to-point error, and feature-space distance. Distribution-consistency metrics assess whether the distributions of the real and generated data are aligned. Point-to-point error metrics focus on the numerical differences between the real and generated samples. When the generated objects are represented as vectors, images, or embeddings, feature-space distance metrics are used to measure their closeness in the representation space. Metrics such as Fréchet inception distance (FID) are usually adopted when traffic is converted into images or evaluated in a learned feature space. In such settings, FID can reflect whether the real and generated samples are close in the transformed representation space. However, this closeness does not by itself imply preserved protocol semantics, valid state transitions, or executable communication behavior. Therefore, FID should be interpreted as a transformed-space similarity metric. Table 4 summarizes the similarity metrics used in network traffic generation studies.

5.2 Usability metrics

In usability evaluation, generated data are assessed through downstream tasks. Classifiers are trained on generated

Table 4 Similarity metrics used in network traffic generation studies

Evaluation path	Metric	Description
Distribution consistency	JSD (Hui et al., 2022)	Measures the symmetric and bounded divergence between the distributions of y and $\hat{y}$
	TVD (Zhang SY et al., 2024)	Measures the statistical distance between the probability distributions of y and $\hat{y}$
	Wasserstein distance (Manocchio et al., 2021)	Measures the minimal transport cost between the probability distributions of y and $\hat{y}$
	CRPS (Chai et al., 2024)	Measures the accuracy of probabilistic forecasts by comparing the predicted CDF with the observed value
Point-to-point error	MAE (Li J and Li, 2022)	Average absolute error between the real and generated values: $\frac{1}{T}\sum_{t=1}^{T}\lvert y_t-\hat{y}_t\rvert$
	MSE (Li T et al., 2024)	Average squared error between the real and generated values: $\frac{1}{T}\sum_{t=1}^{T}(y_t-\hat{y}_t)^2$
	RMSE (Chai et al., 2024)	Square root of the mean squared error: $\sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t-\hat{y}_t)^2}$
	MAPE (Li J and Li, 2022)	Average relative error between the real and generated values: $\frac{1}{T}\sum_{t=1}^{T}\frac{\lvert y_t-\hat{y}_t\rvert}{\lvert y_t\rvert}$
Feature-space distance	FID (Cai et al., 2025)	Measures the Fréchet distance between the feature distributions of y and $\hat{y}$ in a pre-trained feature space
	Cosine similarity (Du et al., 2023)	Measures the cosine of the angle between two vectors
	Euclidean distance (Ring et al., 2019)	Measures the Euclidean distance between two vectors in the feature space

The metrics in this table are synthesized from the reviewed studies summarized in Table 1. Since a given metric may appear in multiple reviewed studies, only a few illustrative references are listed here for brevity rather than citing all relevant works listed in Table 1. JSD: Jensen–Shannon divergence; TVD: total variation distance; CRPS: continuous ranked probability score; MAE: mean absolute error; MSE: mean squared error; RMSE: root mean squared error; MAPE: mean absolute percentage error; CDF: cumulative distribution function

data, or on mixed real and generated data, and then tested on a real test set. Another setting uses classifiers trained on real data to identify generated samples. This helps examine whether the generated samples contain discriminative features that downstream classifiers can recognize. In both settings, metrics such as accuracy, precision, recall, and F1-score are often used to measure the effect of the generated data on classification performance (Zhu et al., 2022; Wang MX et al., 2024; Shin et al., 2025).

In addition, as shown in Table 5, several methods evaluate the generated data from a perspective closer to real deployment. This directly reflects their protocol consistency in practical environments. Specifically, a protocol compliance check examines whether the generated traffic satisfies protocol-related constraints, such as valid field formats, length consistency, checksum correctness, and parser acceptance. Packet success rate measures the proportion of generated packets or traces that can be successfully processed, transmitted, or replayed by real tools, protocol parsers, or network stacks. These metrics provide stronger evidence of practical protocol consistency than similarity scores alone.

Table 6 summarizes the evaluation metrics reported in the studies according to Table 1. L1 remains the closest to real protocol processes, so its evaluation focuses more on protocol validity, structural correctness, and practical usability. L2 still preserves part of the communication structure and is often evaluated from structural and statistical perspectives. L3 is more abstract and is mainly assessed by quantitative similarity and task-consistency results. L4 is the most abstract level and is usually evaluated in terms of abstract similarity and task consistency for downstream applications. Overall, lower-level representations are evaluated more from the perspective of practical applicability, whereas higher-level representations are evaluated more from the perspective of abstraction and task support. Therefore, evaluation results should always be interpreted together with the corresponding representation level.

A practical evaluation should not rely on a single metric. At least one similarity metric and one usability-oriented metric should be considered together. Similarity metrics help assess whether the generated traffic remains statistically close to the real traffic, while usability-oriented metrics examine whether the generated traffic satisfies task consistency or protocol consistency. This combined strategy reduces the risk of misleading conclusions that may arise when a method performs well on only one narrow evaluation criterion.

6 Challenges and directions

As shown in Table 1, among the 39 studies, only 14 studies report explicit protocol-level evidence. Most existing methods are still evaluated mainly by downstream task performance or statistical similarity. In contrast, explicit evidence of replayability, protocol validity, and stateful communication behavior remains limited. This suggests that current methods can often generate traffic data for classification, detection, or distribution matching, but they do not always generate traffic that can be parsed, replayed, or executed in realistic network environments.

This limitation affects practical use and research comparability. In practice, the generated traffic should not only resemble the real traffic statistically, but also meet scenario requirements, follow protocol rules, and behave consistently during communication. For evaluation, isolated metrics such as classification accuracy or distributional similarity are not sufficient to determine whether a generation method can be used in simulation, testing, or security evaluation. Therefore, future work should study traffic generation from a broader perspective that connects controllable task requirements, protocol-aware synthesis, and reproducible engineering validation.

To summarize these challenges more clearly, this section discusses three future research directions. As shown in Fig. 5, these directions are task-oriented generation, protocol-oriented generation, and engineering-oriented evaluation.

6.1 Task-oriented generation

As network traffic generation is increasingly considered in practical settings, controllability has received growing attention. Many existing methods focus on matching the overall data distribution. However, this does not always meet the needs of scenario-specific generation. In many applications, researchers require traffic under particular conditions rather than traffic that only reflects general patterns. Examples

Table 5 Usability metrics used in network traffic generation studies

Usability metric	Description
ASR (Sun PS et al., 2025)	Measures the proportion of the generated adversarial samples that successfully bypass the target detection model, reflecting the attack effectiveness of $\hat{y}$
Protocol compliance check (Sun PS et al., 2025)	Examines whether the generated packets satisfy protocol rules, such as field constraints, packet length, checksum validity, and state transition logic
Domain knowledge check (Ring et al., 2019)	Examines whether $\hat{y}$ is consistent with basic network protocol rules and the characteristics of the target dataset
Packet success rate (Delgado-Soto et al., 2025)	Measures the proportion of the generated packets that are transmitted correctly without errors. It reflects whether the generated packets can be processed by network devices

The metrics in this table are synthesized from the reviewed studies summarized in Table 1. Since a given metric may appear in multiple reviewed studies, only a few illustrative references are listed here for brevity rather than citing all relevant works listed in Table 1. ASR: attack success rate

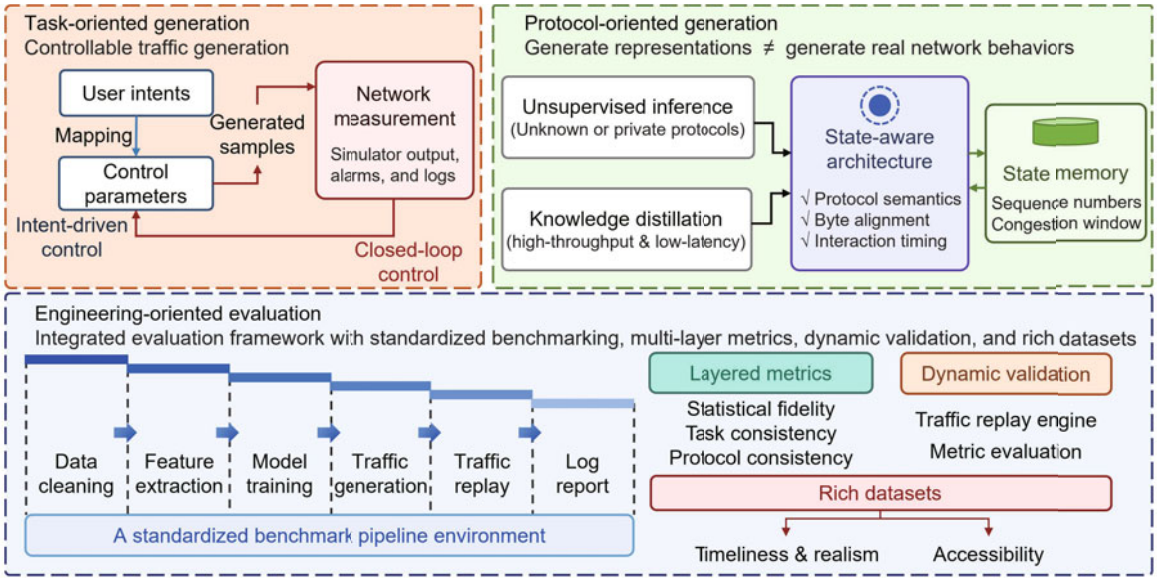

Fig. 5 Future directions of network traffic generation

Table 6 Common evaluation metrics across different traffic representation levels

Metric	L1	L2	L3	L4
JSD	×	✓	✓	×
TVD	×	×	✓	×
Wasserstein distance	✓	✓	✓	✓
CRPS	×	×	✓	×
MAE	×	✓	✓	×
MSE	×	✓	✓	×
RMSE	×	✓	✓	×
MAPE	×	×	✓	×
FID	×	×	✓	✓
Cosine similarity	×	✓	✓	✓
Euclidean distance	×	✓	✓	✓
ASR	✓	×	×	×
Protocol compliance check	✓	×	×	×
Domain knowledge check	✓	×	×	×
Packet success rate	✓	×	×	×

✓ indicates metrics typically used at the level; × indicates metrics not typically used at the level; JSD: Jensen–Shannon divergence; TVD: total variation distance; CRPS: continuous ranked probability score; MAE: mean absolute error; MSE: mean squared error; RMSE: root mean squared error; MAPE: mean absolute percentage error; ASR: attack success rate

include bursty DDoS attacks, abnormal flows under specific topologies, and rare failure events. When control over the generated output is limited, the value for practical testing and evaluation of these methods is limited.

This issue is closely related to real application needs. In tasks such as intrusion detection evaluation and attack–defense experiments, users often need to adjust factors such as attack intensity, background traffic patterns, and node behavior. If a model cannot represent these conditions, its use in scenario-based testing becomes restricted. For this reason, task-oriented traffic generation is concerned not only with realism, but also with the ability to specify generation conditions in a clear and usable way. These conditions may be defined at different levels, including high-level scenarios and low-level traffic attributes.

Some recent studies have already explored early forms of controllable traffic generation. For example, Poisson et al. (2024) supported controllable IoT traffic generation through configuration files. Users can specify parameters such as the number of sensors, message rate, attack intensity, and the proportion of compromised devices. Based on these settings, the system generates benign and malicious traffic for services such as message queuing telemetry transport (MQTT) and Kafka, and provides automatic labels. In addition, LLM-based frameworks use staged correction and controllable decoding to achieve more flexible generation (Huang et al., 2026). However, these methods rely mainly on manually specified conditions or local repair mechanisms and do not yet adapt generation in response to feedback from the target environment.

Future research may extend traffic generation beyond fixed parameter settings. One possible direction is intent-level control. In this setting, high-level scenario descriptions are linked to executable generation conditions. For example, a target scenario may be expressed through factors such as traffic rate, packet interval, TCP behavior, and source–destination distribution. This form of control may provide a more direct way to describe testing requirements and may reduce reliance on manual low-level parameter design. Intent-level control may be implemented through a two-stage framework. In the first stage, a high-level user intent is translated into structured generation constraints, such as protocol type, traffic rate, packet interval, and attack intensity. This translation can be supported by rule-based parsers, templates, or LLM-based semantic parsers. In the second stage, these constraints are used as conditions for generators, such as conditional GANs or conditional DMs.

Another possible direction is closed-loop control. In many practical settings, traffic generation is not a one-time process. Instead, it may change according to feedback during execution.

Such feedback may come from intrusion detection system (IDS) alarms, simulator outputs, or network state measurements. Under this setting, the generation process can be updated over time. This is relevant to dynamic tasks such as robustness testing, attack–defense experiments, and automated system validation. Closed-loop control may be implemented by connecting the generator with an external feedback source, such as an IDS, a simulator, or a network emulator. The generator first produces traffic under given conditions. The environment then returns feedback, such as the detection rate, replay success rate, or packet loss. Based on this feedback, the generation policy can be updated through reinforcement learning. This design would allow traffic generation to adapt to dynamic testing objectives rather than producing fixed offline samples.

Overall, task-oriented generation provides a useful perspective for extending network traffic generation beyond passive distribution matching. Compared with methods that focus only on reproducing overall traffic patterns, it places more attention on scenario requirements and operational conditions. This perspective is closely related to the use of generated traffic in practical testing and evaluation.

6.2 Protocol-oriented generation

A common limitation in the current network traffic generation research is that generated samples do not always correspond to the real network behavior. Many existing methods are designed for a specific representation space, such as aggregated flow statistics or raw hexadecimal sequences. These representations may be sufficient for fitting data distributions or supporting downstream classifiers. However, they often fail to preserve protocol semantics, interaction states, and contextual dependencies that are important for real network communication. As a result, generated traffic may appear realistic in a statistical sense, while still being difficult to interpret, replay, or transfer across scenarios.

Existing studies have made early attempts to improve protocol consistency. For example, Delgado-Soto et al. (2025) and Huang et al. (2026) introduced protocol-aware generation, constrained decoding, or repair mechanisms to reduce malformed outputs and improve replayability. These studies show that protocol consistency can be improved when generation is guided by the protocol structure rather than unconstrained sequence modeling alone. However, current approaches remain limited. GAN- and DM-based methods often struggle to preserve byte-level structural validity and require filtering or repair, which may affect reproducibility and cross-scenario consistency. LLM-based methods provide a more flexible way to condition generation through natural language, but their tokenization and decoding processes are not naturally aligned with packet bytes or protocol fields (Huang et al., 2026). More importantly, most methods focus on local packet correctness or post-generation repair, while offering limited support for timing dependencies, state transitions, and session-level consistency across multi-packet interactions. As a result, their use in the real network environments remains constrained, especially when the generated traffic must be replayed, interpreted, or validated with tools such as Tcpreplay and Wireshark (Jiang et al., 2024).

These observations suggest that future research may need to move beyond representation-oriented synthesis toward protocol-oriented generation. In this setting, the goal is not only to assign valid values to packet fields, but also to model the syntactic and semantic rules that govern network communication. Generated traffic should preserve valid field structures, reasonable timing behavior, and consistent protocol state transitions. This would create a connection between generated traffic and real protocol behavior.

From this perspective, one possible direction is state-aware generation. Existing methods often focus on local packet correctness or short-range sequence patterns. However, real network traffic is inherently stateful. Long-lived connections involve dynamic variables such as sequence numbers, congestion window changes, retransmission behavior, and encrypted handshake context. Future models may benefit from explicit protocol state memory or state transition modeling, so that protocol behavior remains coherent across complete sessions rather than isolated packets. State-aware generation can be realized by combining the generator with a protocol-state constraint layer. For TCP traffic, the model may maintain a session-level state machine with states such as SYN-SENT, ESTABLISHED, FIN-WAIT, and CLOSED. During decoding, the generator proposes the next packet or field values, while the constraint layer checks whether the proposed output is valid under the current TCP state. Invalid transitions, such as transmitting data before connection establishment or producing inconsistent FIN/ACK behavior, can then be blocked or corrected before output. In addition, sequence numbers, acknowledgment numbers, retransmission behavior, and timing constraints can be tracked as state variables throughout the session.

Another direction is traffic generation for unknown or private protocols. Current protocol-aware approaches often rely on requests for comments (RFCs) (Poisson et al., 2024), manually designed templates, or prompt engineering. This limits their use when formal protocol specifications are unavailable. One possible approach is unsupervised protocol inference, where the model learns protocol grammar, field constraints, and interaction patterns directly from raw traffic. If the inferred structure can be transformed into executable generation logic, such methods may provide a basis for generating traffic under undocumented, private, or previously unseen protocols.

A further direction concerns practical deployment constraints. Large models may provide richer protocol knowledge, but inference latency and throughput can limit their use in high-speed or real-time settings. Knowledge distillation offers one possible solution. Protocol knowledge learned by large models may be transferred to lighter generators, improving efficiency while retaining key structural information.

Overall, protocol-oriented generation provides an important perspective for future research. Compared with methods that mainly focus on feature-level similarity, it places greater emphasis on communication behavior, state transitions, and protocol validity. This perspective is closely related to the use of generated traffic in simulation, testing, and security evaluation.

6.3 Engineering-oriented evaluation

Current evaluation practices in network traffic generation research show several common limitations. One major issue is the lack of unified evaluation standards, together with large variation in experimental environments across studies. Many methods are assessed mainly by statistical similarity or downstream task performance. These results do not cover all important properties of the generated traffic. In particular, protocol compliance, interaction consistency, and cross-scenario generalization are not always examined systematically. Consequently, direct comparison across methods remains difficult, and the practical applicability of the generated traffic is always unclear. Reproducibility is another important issue. Existing studies often differ in data preprocessing, protocol parsing tools, software versions, and experimental settings. Common benchmark datasets and fixed test topologies are limited. Therefore, the results are often difficult to reproduce across research groups or transfer to other environments.

In this regard, a possible direction is the construction of standardized benchmark environments, which can reduce implementation and environmental variations. Containerized pipelines can organize data cleaning, feature extraction, model training, traffic generation, traffic replay, and result reporting into a repeatable workflow with fixed software dependencies. Furthermore, benchmark-defined network topologies, software versions, key hyperparameters, and random seeds should be documented. This approach would make experimental results easier to reproduce and compare across different methods and research groups.

Another direction is the development of a layered metric system. Evaluation should not rely on a single perspective but should instead cover at least three layers: statistical fidelity, task consistency, and protocol consistency. Statistical fidelity measures whether the generated traffic matches the real traffic in terms of packet length, inter-arrival time, and payload-related distributions. Task consistency evaluates whether the generated traffic remains useful for downstream applications, such as intrusion detection, traffic classification, and robustness evaluation. Protocol consistency examines whether the generated traffic follows protocol rules and communication logic through protocol parsing, field validation, and session interaction checks. Incorporating these metrics into a standard benchmark would provide a broader and more comparable basis for evaluating different traffic generation methods.

Existing approaches primarily rely on offline analysis. Such methods cannot fully capture how the generated traffic behaves in interactive network environments. Therefore, dynamic validation should be introduced as a necessary complement to offline evaluation. Virtual network stacks, emulators, or controlled testbeds can be used to replay the generated traffic under conditions such as packet loss, congestion, and routing changes. Researchers can then observe protocol behavior and system responses during execution. Dynamic metrics, such as replay success rate, abnormal retransmission rate, and session completion rate, can be used to assess whether the generated traffic remains valid during interaction. Such validation would bring evaluation closer to practical deployment settings.

Besides metric design, benchmark quality also depends on the datasets used for evaluation. In current studies, many methods are tested on a small set of older public datasets, while some newer datasets are private or self-constructed. This dataset gap further affects comparability, reproducibility, and the practical meaning of the reported results. Table 7 summarizes the datasets used by representative traffic generation studies.

Table 7 reveals two issues regarding datasets. First, several widely reused public datasets were released many years ago. Their accessibility and established usage make them useful as reference benchmarks, but they may no longer fully represent the current application behavior, encryption deployment, or IoT communication patterns. Second, several recent studies rely on operator data, self-collected traces, or self-constructed datasets that are not publicly available. While these datasets may better reflect current traffic, they are usually unavailable due to privacy or sharing restrictions, making fair comparison more difficult and weakening reproducibility. Therefore, engineering-oriented evaluation should consider not only metrics and replay settings but also dataset age, accessibility, and realism.

Ideally, engineering-oriented evaluation should combine standardized benchmarks, multi-layer metrics, dynamic validation, and rich datasets. This integrated perspective would provide stronger evidence regarding whether the generated traffic is statistically realistic, useful for downstream tasks, protocol-consistent, and applicable in practical network environments.

7 Conclusions

The central finding of this survey is that traffic suitable for downstream machine learning tasks is not necessarily the traffic that can be replayed, executed, or validated as protocol-consistent network communication. This indicates that network traffic generation should not be judged only by whether generated samples improve classification, detection, or prediction performance. It also requires attention to what form of traffic is generated and what practical objective the generated traffic can support. Therefore, this review organizes network traffic generation from two connected perspectives: representation levels and objective consistency. The representation-level framework clarifies what form of traffic is generated, from L1 to L4. The objective consistency framework clarifies what the generated traffic is expected to support, especially the distinction between task consistency and protocol consistency. Together, these two perspectives reveal a clear trade-off. Low-level representations preserve more protocol semantics, packet dependencies, and replayability, but they are more challenging to learn and generate accurately. High-level representations are easier for generative models to fit, but they often remove the communication logic required for protocol-aware use.

From a practical perspective, the choice of representation level should depend on the intended use of the generated traffic. If the goal is protocol testing, traffic replay, security simulation, or executable PCAP generation, L1 is the most suitable choice, as it preserves packet fields and protocol structure. In this setting, evaluation should include protocol compliance checks, parser acceptance, packet success rate, replay success rate, invalid packet ratio, and session-level state validation.

Table 7 Traffic datasets used by representative traffic generation studies

Dataset	Year	Traffic type	Main limitation	Study
NSL-KDD (Tavallaee et al., 2009)	2009	Flow-level IDS features	Feature-based	Charlier et al. (2019)
UNSW-NB15 (Moustafa and Slay, 2015)	2015	Benign and attack enterprise traffic	Testbed characteristics	Wang MX et al. (2024)
ISCX VPN-nonVPN (Draper-Gil et al., 2016)	2016	VPN and non-VPN application traffic	Limited class scale	Shin et al. (2025)
CICIDS2017 (Sharafaldin et al., 2018)	2017	Enterprise traffic with benign and multiple attack classes	Known labeling and preprocessing issues	Charlier et al. (2019); Mozo et al. (2022); Wang MX et al. (2024); Cai et al. (2025)
Bot-IoT (Koroniotis et al., 2019)	2018	IoT botnet and attack traffic	Attack-centric	Huang et al. (2026)
CSE-CIC-IDS2018 (Sharafaldin et al., 2018)	2018	Enterprise and attack traffic	Testbed characteristics	Wang MX et al. (2024); Cai et al. (2025); Jonath et al. (2026)
CIC-DDoS2019 (Sharafaldin et al., 2019)	2019	DDoS attack traffic	Narrow attack focus	Cai et al. (2025)
TON_IoT (Alsaedi et al., 2020)	2019	IoT and attack traffic	Strong IoT specificity	Yin et al. (2022); Delgado-Soto et al. (2025); Huang et al. (2026); Jonath et al. (2026)
User traffic dataset	2016	Mobile user traffic	Not public	Li T et al. (2024)
VarLen-AnyDom, FixLen-OneDom, and FixLen-TwoDom	2022	DNS packet traffic	Self-constructed	Nukavarapu et al. (2022)
Beijing, Shanghai, Nanjing, Nanchang, and Shandong mobile traffic datasets	2024–2025	Mobile traffic	Not public	Chai et al. (2024, 2025a); Qi et al. (2024)

VPN: virtual private network

If the goal is traffic evolution modeling, workload simulation, or sequence-level behavior analysis, L2 is more appropriate, as it retains packet or flow order, timing, and communication patterns. In this case, evaluation should combine temporal distribution metrics, such as packet length and inter-arrival time distributions, with task-based metrics when downstream applications are involved. If the goal is intrusion detection, traffic classification, prediction, or data augmentation, L3 and L4 are sufficient, as they provide compact and learnable feature spaces. Their evaluation should focus on statistical fidelity, feature-space distances, and downstream performance metrics such as accuracy, precision, recall, and F1-score. However, good performance at L3 or L4 should not be treated as evidence of replayability or protocol consistency.

Overall, future progress in network traffic generation will depend not only on stronger generative models, but also on a clearer alignment among representation level, generation objective, and evaluation strategy. The generated traffic should therefore be judged not only by whether it improves downstream task performance, but also by whether it supports the practical applications for which it is intended. This alignment is essential for advancing the field from task-oriented data augmentation toward more reliable, protocol-aware, and practically usable traffic generation.

Author contributions

Biying WANG designed the framework, conducted the literature analysis, and drafted the paper. Shuang ZHAO, Minxin WANG, and Zhengpeng LI assisted in literature collection and paper organization. Jinshu SU, Baosheng WANG, Shuhui CHEN, Shuang ZHAO, and Ziling WEI supervised the review. Biying WANG, Ziling WEI, and Shuang ZHAO revised the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Declaration on the use of generative AI tools

During the preparation of this work, the authors used DeepSeek solely for language polishing and improving readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

Adeleke OA, Bastin N, Gurkan D, 2023. Network traffic generation: a survey and methodology. *ACM Comput Surv*, 55(2):28. https://doi.org/10.1145/3488375

Alsaedi A, Moustafa N, Tari Z, et al., 2020. TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, 8:165130-165150. https://doi.org/10.1109/ACCESS.2020.3022862

Cai SH, Zhao XY, Chen JF, et al., 2025. CT-SSSA: malicious traffic augmentation based on classifier TransGAN and spatial-channel synergistic self-attention. *Knowl-Based Syst*, 329:114285. https://doi.org/10.1016/j.knosys.2025.114285

Carillo R, Cerasuolo F, Bovenzi G, et al., 2025. Explainable federated class incremental learning for encrypted network traffic classification. *Comput Netw*, 269:111448. https://doi.org/10.1016/j.comnet.2025.111448

Chai HY, Jiang T, Yu L, 2024. Diffusion model-based mobile traffic generation with open data for network planning and optimization. Proc 30th Conf on Knowledge Discovery and Data Mining, p.4828-4838. https://doi.org/10.1145/3637528.3671544

Chai HY, Qi XQ, Li Y, 2025a. Spatio-temporal knowledge driven diffusion model for mobile traffic generation. *IEEE Trans Mob Comput*, 24(6):4939-4956. https://doi.org/10.1109/TMC.2025.3527966

Chai HY, Zhang SY, Qi XQ, et al., 2025b. UoMo: a universal model of mobile traffic forecasting for wireless network optimization. Proc 31st Conf on Knowledge Discovery and Data Mining V.2, p.4308-4319. https://doi.org/10.1145/3711896.3737272

Charlier J, Singh A, Ormazabal G, et al., 2019. SynGAN: towards generating synthetic network attacks using GANs. https://doi.org/10.48550/arXiv.1908.09899

Chawla NV, Bowyer KW, Hall LO, et al., 2002. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res*, 16(1):321-357. https://doi.org/10.1613/jair.953

Chen YC, 2017. A tutorial on kernel density estimation and recent advances. *Biostat Epidemiol*, 1(1):161-187. https://doi.org/10.1080/24709360.2017.1396742

Cheng A, 2019. PAC-GAN: packet generation of network traffic using generative adversarial networks. IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conf, p.728-734. https://doi.org/10.1109/IEMCON.2019.8936224

Chu A, Jiang X, Liu SN, et al., 2024. Feasibility of state space models for network traffic generation. Proc SIGCOMM Workshop on Networks for AI Computing, p.9-17. https://doi.org/10.1145/3672198.3673792

Chu A, Jiang X, Liu SN, et al., 2026. NetSSM: multi-flow and state-aware network trace generation using state-space models. *Proc ACM Netw*, 4(CoNEXT1):6. https://doi.org/10.1145/3786289

Delgado-Soto JA, de Vergara JEL, González I, et al., 2025. GPT on the wire: towards realistic network traffic conversations generated with large language models. *Comput Netw*, 265:111308. https://doi.org/10.1016/j.comnet.2025.111308

Dowoo B, Jung Y, Choi C, 2019. PcapGAN: packet capture file generator by style-based generative adversarial networks. 18th IEEE Int Conf on Machine Learning and Applications, p.1149-1154. https://doi.org/10.1109/ICMLA.2019.00191

Draper-Gil G, Lashkari AH, Mamun MSI, et al., 2016. Characterization of encrypted and VPN traffic using time-related features. Proc 2nd Int Conf on Information Systems Security and Privacy, p.407-414. https://doi.org/10.5220/0005740704070414

Du LF, He JJ, Li T, et al., 2023. DBWE-Corbat: background network traffic generation using dynamic word embedding and contrastive learning for cyber range. *Comput Secur*, 129:103202. https://doi.org/10.1016/j.cose.2023.103202

Fernandes DAB, Neto M, Soares LFB, et al., 2015. On the self-similarity of traffic generated by network traffic simulators. In: Obaidat MS, Nicopolitidis P, Zarai F (Eds.), Modeling and Simulation of Computer Networks and Systems. Morgan Kaufmann, Waltham, p.285-311. https://doi.org/10.1016/B978-0-12-800887-4.00010-9

Goodfellow IJ, Pouget-Abadie J, Mirza M, et al., 2014. Generative adversarial networks. Proc 28th Int Conf on Neural Information Processing Systems, p.2672-2680. https://dl.acm.org/doi/10.5555/2969033.2969125

Ho J, Jain A, Abbeel P, 2020. Denoising diffusion probabilistic models. Proc 34th Int Conf on Neural Information Processing Systems, Article 574. https://dl.acm.org/doi/abs/10.5555/3495724.3496298

Hochreiter S, Schmidhuber J, 1997. Long short-term memory. *Neur Comput*, 9(8):1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang YZ, Li XH, Geng JX, et al., 2026. TrafficT5: multi-stage self-correcting framework for traffic generation. *Comput Netw*, 274:111858. https://doi.org/10.1016/j.comnet.2025.111858

Hui SD, Wang HD, Wang ZH, et al., 2022. Knowledge enhanced GAN for IoT traffic generation. Proc ACM Web Conf, p.3336-3346. https://doi.org/10.1145/3485447.3511976

Jablaoui R, Liouane N, 2026. GA-CNN-BiGRU-IDS: a robust framework for intrusion detection system based on GA for data augmentation and hybrid CNN-BiGRU model for spatiotemporal feature extraction. *Comput Electr Eng*, 130:110900. https://doi.org/10.1016/j.compeleceng.2025.110900

Jiang X, Liu SN, Gember-Jacobson A, et al., 2024. NetDiffusion: network data augmentation through protocol-constrained traffic generation. *Proc ACM Meas Anal Comput Syst*, 8(1):11. https://doi.org/10.1145/3639037

Jonath KK, Naz A, Zhang SG, 2026. GMM-cGAN: mitigating data scarcity and label noise for robust encrypted malicious traffic classification. *Comput Netw*, 274:111823. https://doi.org/10.1016/j.comnet.2025.111823

Kattadige C, Muramudalige SR, Choi KN, et al., 2021. VideoTrain: a generative adversarial framework for synthetic video traffic generation. IEEE 22nd Int Symp on a World of Wireless, Mobile and Multimedia Networks, p.209-218. https://doi.org/10.1109/WoWMoM51794.2021.00034

Kennedy J, Eberhart R, 1995. Particle swarm optimization. Proc Int Conf on Neural Networks, p.1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Kholgh DK, Kostakos P, 2023. PAC-GPT: a novel approach to generating synthetic network traffic with GPT-3. *IEEE Access*, 11:114936-114951. https://doi.org/10.1109/ACCESS.2023.3325727

Kingma DP, Welling M, 2014. Auto-encoding variational Bayes. https://doi.org/10.48550/arXiv.1312.6114

Koroniotis N, Moustafa N, Sitnikova E, et al., 2019. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gener Comput Syst*, 100:779-796. https://doi.org/10.1016/j.future.2019.05.041

Li J, Li X, 2022. 5G network traffic prediction based on EEMD-GAN. Proc 7th Int Conf on Cyber Security and Information Engineering, p.408-412. https://doi.org/10.1145/3558819.3565116

Li T, Hui SD, Zhang SY, et al., 2024. Mobile user traffic generation via multi-scale hierarchical GAN. *ACM Trans Knowl Discov Data*, 18(8):189. https://doi.org/10.1145/3664655

Lin ZN, Jain A, Wang C, et al., 2020. Using GANs for sharing networked time series data: challenges, initial promise, and open questions. Proc ACM Int Measurement Conf, p.464-483. https://doi.org/10.1145/3419394.3423643

Manocchio LD, Layeghy S, Portmann M, 2021. FlowGAN - synthetic network flow generation using generative adversarial networks. IEEE 24th Int Conf on Computational Science and Engineering, p.168-176. https://doi.org/10.1109/CSE53436.2021.00033

Meddahi A, Drira H, Meddahi A, 2021. SIP-GAN: generative adversarial networks for SIP traffic generation. Int Symp on Networks, Computers and Communications, p.1-6. https://doi.org/10.1109/ISNCC52172.2021.9615632

Meslet-Millet F, Mouysset S, Chaput E, 2022. NeCSTGen: an approach for realistic network traffic generation using deep learning. IEEE Global Communications Conf, p.3108-3113. https://doi.org/10.1109/GLOBECOM48099.2022.10000731

Meta AI, 2024. Llama3/MODEL_CARD.md. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md [Accessed on May 25, 2026].

Moustafa N, Slay J, 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). Military Communications and Information Systems Conf, p.1-6. https://doi.org/10.1109/MilCIS.2015.7348942

Mozo A, González-Prieto Á, Pastor A, et al., 2022. Synthetic flow-based cryptomining attack generation through generative adversarial networks. *Sci Rep*, 12(1):2091. https://doi.org/10.1038/s41598-022-06057-2

Nukavarapu SK, Ayyat M, Nadeem T, 2022. MirageNet - towards a GAN-based framework for synthetic network traffic generation. IEEE Global Communications Conf, p.3089-3095. https://doi.org/10.1109/GLOBECOM48099.2022.10001494

Oh S, Oh MJ, Im JK, et al., 2025. Spatio-temporal data augmentation method for network traffic prediction. *IEEE Access*, 13:138686-138698. https://doi.org/10.1109/ACCESS.2025.3595120

Poisson M, Carnier RM, Fukuda K, 2024. GothX: a generator of customizable, legitimate and malicious IoT network traffic. Proc 17th Cyber Security Experimentation and Test Workshop, p.65-73. https://doi.org/10.1145/3675741.3675753

Qi XQ, Chai HY, Yu L, et al., 2024. Regional features conditioned diffusion models for 5G network traffic generation. Proc 32nd ACM Int Conf on Advances in Geographic Information Systems, p.396-409. https://doi.org/10.1145/3678717.3691312

Ring M, Schlör D, Landes D, et al., 2019. Flow-based network traffic generation using generative adversarial networks. *Comput Secur*, 82:156-172. https://doi.org/10.1016/j.cose.2018.12.012

Shafi M, Lashkari AH, Roudsari AH, 2025. NTLFlowLyzer: towards generating an intrusion detection dataset and intruders behavior profiling through network and transport layers traffic analysis and pattern extraction. *Comput Secur*, 148:104160. https://doi.org/10.1016/j.cose.2024.104160

Shahid MR, Blanc G, Jmila H, et al., 2020. Generative deep learning for Internet of Things network traffic generation. IEEE 25th Pacific Rim Int Symp on Dependable Computing, p.70-79. https://doi.org/10.1109/PRDC50213.2020.00018

Shapira T, Shavitt Y, 2021. FlowPic: a generic representation for encrypted traffic classification and applications identification. *IEEE Trans Netw Serv Manage*, 18(2):1218-1232. https://doi.org/10.1109/TNSM.2021.3071441

Sharafaldin I, Habibi Lashkari A, Ghorbani AA, 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proc 4th Int Conf on Information Systems Security and Privacy, p.108-116. https://doi.org/10.5220/0006639801080116

Sharafaldin I, Lashkari AH, Hakak S, et al., 2019. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. Int Carnahan Conf on Security Technology, p.1-8. https://doi.org/10.1109/CCST.2019.8888419

Shawkat M, Badawi M, El-ghamrawy S, et al., 2022. An optimized FP-growth algorithm for discovery of association rules. *J Supercomput*, 78:5479-5506. https://doi.org/10.1007/s11227-021-04066-y

Shin CY, Choi YS, Kim MS, 2025. Data augmentation-based enhancement for efficient network traffic classification. *IEEE Access*, 13:6006-6028. https://doi.org/10.1109/ACCESS.2024.3525000

Sivaroopan N, Bandara D, Madarasingha C, et al., 2024. NetDiffus: network traffic generation by diffusion models through time-series imaging. *Comput Netw*, 251:110616. https://doi.org/10.1016/j.comnet.2024.110616

Sivaroopan N, Silva K, Madarasingha C, et al., 2026. A comprehensive survey on synthetic network traffic generation. *IEEE Commun Surv Tutor*, 28:5949-5983. https://doi.org/10.1109/COMST.2026.3684111

Soper J, Xu Y, Foo E, et al., 2024. Improved packet-level synthetic network traffic generation. IEEE 23rd Int Conf on Trust, Security and Privacy in Computing and Communications, p.1928-1934. https://doi.org/10.1109/TrustCom63139.2024.00267

Sun DY, Chen JQ, Gong C, et al., 2024. NetDPSyn: synthesizing network traces under differential privacy. Proc ACM on Internet Measurement Conf, p.545-554. https://doi.org/10.1145/3646547.3689011

Sun PS, Yun XC, Li SH, et al., 2025. AdvTG: an adversarial traffic generation framework to deceive DL-based malicious traffic detection models. Proc ACM on Web Conf, p.3147-3159. https://doi.org/10.1145/3696410.3714876

Tavallaee M, Bagheri E, Lu W, et al., 2009. A detailed analysis of the KDD CUP 99 data set. IEEE Symp on Computational Intelligence for Security and Defense Applications, p.1-6. https://doi.org/10.1109/CISDA.2009.5356528

Wang BY, Wang BS, Wei ZL, et al., 2025. MFSI: multi-flow based service identification for encrypted network traffic. *Comput Netw*, 265:111283. https://doi.org/10.1016/j.comnet.2025.111283

Wang MX, Yang N, Forcade-Perkins NJ, et al., 2024. ProGen: projection-based adversarial attack generation against network intrusion detection. *IEEE Trans Inform Forensics Secur*, 19:5476-5491. https://doi.org/10.1109/TIFS.2024.3402155

Yang LM, Wang YJ, Liu L, et al., 2025. unFlowS: an unsupervised construction scheme of flow spectrum for network traffic detection. *IEEE Trans Inform Forensics Secur*, 20:3330-3345. https://doi.org/10.1109/TIFS.2025.3550060

Yin YC, Lin Z, Jin M, et al., 2022. Practical GAN-based synthetic IP header trace generation using NetShare. Proc ACM SIGCOMM Conf, p.458-472. https://doi.org/10.1145/3544216.3544251

Zhang HZ, 2024. TransFlowGAN: generation and balancing of network traffic data. Proc Asia Pacific Conf on Computing Technologies, Communications and Networking, p.51-59. https://doi.org/10.1145/3685767.3685776

Zhang JH, Tang JQ, Zhang X, et al., 2015. A survey of network traffic generation. 3rd Int Conf on Cyberspace Technology, p.1-6. https://doi.org/10.1049/cp.2015.0862

Zhang S, Azizi S, Joshi A, et al., 2025. Towards behavior grammar-driven IoT network traffic generation using MUD specifications. Proc 7th Joint Workshop on CPS & IoT Security and Privacy, p.98-104. https://doi.org/10.1145/3733801.3764203

Zhang SY, Li T, Jin DP, et al., 2024. NetDiff: a service-guided hierarchical diffusion model for network flow trace generation. *Proc ACM Netw*, 2(CoNEXT3):16. https://doi.org/10.1145/3676870

Zhu XF, Shu NN, Zheng BW, et al., 2022. A method to generate a ground truth distributed network traffic dataset. Proc 3rd Int Conf on Control, Robotics and Intelligent System, p.219-224. https://doi.org/10.1145/3562007.3562049

Zion Y, Aharon P, Dubin R, et al., 2025. Enhancing encrypted Internet traffic classification through advanced data augmentation techniques. IEEE Int Conf on Communications, p.1-6. https://doi.org/10.1109/ICC52391.2025.11160885