AI-based Traffic Modeling for Network Security and Privacy: Challenges Ahead

Dinil Mon Divakaran

A*STAR Institute for Infocomm Research (A*STAR I^2R)
dinil_divakaran@a-star.edu.sg

Abstract

Network traffic analysis using AI (machine learning and deep learning) models made significant progress over the past decades. Traffic analysis addresses various challenging problems in network security, ranging from detection of anomalies and attacks to countering of Internet censorship. AI models are also developed to expose user privacy risks as demonstrated by the research works on fingerprinting of user-visiting websites, IoT devices, and different applications, even when payloads are encrypted.

Despite these advancements, significant challenges remain in the domain of network traffic analysis to effectively secure our networks from evolving threats and attacks. After briefly reviewing the relevant tasks and recent AI models for traffic analysis, we discuss the challenges that lie ahead.

1 Introduction

The sophistication of networks attacks has increased over time due to the rapid proliferation of new technologies, applications, network protocols and devices. Meanwhile, network traffic keeps increasing, both in volume and speed, further complicating security challenges. To effectively secure the networks, traffic analysis has long been recognized as a critical component. As payloads in network packets are hardly available for signature-based detection with the high adoption of TLS by different applications (Google 2025a,b), statistical models are becoming inevitable to analyze network traffic. Decades of research efforts have led to the development of various statistical, machine learning (ML) and deep learning (DL) models of different capabilities to tackle multiple network security and privacy (`NetS&P`) tasks.

DL models, in particular, are valuable for network traffic analysis in `NetS&P` tasks for two key reasons, or rather, to address two significant challenges. First, network data is enormous; for example, a 1 Gbps link generates hundreds of gigabytes of data per day under high utilization, even when considering only packet headers (excluding the payload). And network bandwidth has increased steadily, with even consumer (broadband) bandwidth now reaching 1 Gbps. Meanwhile, enterprises and telcos/ISPs utilize bandwidth capacities that are orders of magnitude higher. Second, the number of features that can be extracted from network traces can potentially reach several hundreds or even thousands in count. For instance, when modeling a user browsing session as a data point for training/inference, a small session can easily consist of 200 packets, with each packet containing 10 attributes (e.g., packet size, inter-arrival time, and so on). Consequently, representing this relatively short session requires 2,000 features. As we know, DL models are able to process and learn useful patterns from huge datasets with large feature space.

In light of the current state-of-the-art of AI in network traffic modeling, this work discusses the key challenges that lie ahead in securing networks and protecting user privacy.

2 `NetS&P`: Tasks and Models

We discuss important `NetS&P` tasks and AI-based solutions in the literature, without aiming for exhaustiveness.

2.1 Anomaly detection

Anomalous behaviors occurring on endpoints that manifest in network traffic are the anomalies we aim to detect. In an enterprise network, such anomalies may arise from the adoption of new applications (e.g., ChatGPT), sudden use of new protocols—say, encrypted DNS protocols such as DNS-over-HTTPS (DoH), or DNS-over-TLS (DoT), introduction of new devices (e.g., smartwatches), and so on. At homes or hotel rooms, the presence of hidden IoT devices is a security anomaly (Sharma et al. 2022). Malicious activities such as bot communications, malware traffic and DDoS attacks are also anomalies. For network administrators, these anomalies pose potential threats that must be detected and analyzed to determine appropriate mitigation actions, such as implementing new policies for newly introduced devices or blocking IP addresses associated with identified bots.

AI models: The primary challenge for this task stems from the diverse types of network anomalies, including those that are previously unknown. For instance, while the recent encrypted DNS protocols like DoH provide better privacy for users, they also offer malware developers new ways to hide their communications (Lyu, Gharakheili, and Sivaraman 2022), say, for exfiltrating sensitive information (Ozery, Nadler, and Shabtai 2024). These are relatively new anomalies unknown before DoH was deployed. Existing research in literature tackles this challenge of anomaly detec-

tion by developing unsupervised and semi-supervised models. Early research works explored PCA (Principal Component Analysis), statistical hypothesis testing and regression models to detect anomalous patterns in network traffic (Lakhina, Crovella, and Diot 2004; Brauckhoff, Salamatian, and May 2009; Nevat et al. 2018; Divakaran et al. 2017). With the emergence of generative deep learning models for unsupervised problem settings (Zhou and Paffenroth 2017; Zenati et al. 2018), we witnessed the development of reconstruction-based unsupervised network anomaly detection solutions in the last decade (Mirsky et al. 2018; Nguyen et al. 2019a). For example, GEE (Nguyen et al. 2019a) trains a variational autoencoder (VAE) on noisy benign network traffic sessions to learn its corresponding representation in the latent space, which it subsequently uses to detect deviating behaviors during the inference stage.

2.2 Attack classification

While anomaly detection solutions are useful for detecting broad and unknown types of threats and attacks, they do not identify the specific attack type, which is important to decide on the appropriate mitigation strategy. For example, DDoS traffic must be immediately blocked as close to the sources as possible (a challenge in itself); whereas if a malware is detected, the infected machine has to be contained, incident response initiated, and forensic analysis triggered. Identifying the type of attack is achieved by classifying the network data into one or more *known* attack classes of interest, such as botnet activity, DDoS attacks, C&C communications, and password spraying, among others. Anomaly detection and attack classification typically complement each other, e.g., as demonstrated in (Sudheera et al. 2021), enabling i) the detection of both known and unknown attacks, and ii) the identification of attack types.

AI models: While the problem might appear to be a multi-class classification problem, the data available for different attacks vary. DDoS requires simple features that capture the traffic rate, whereas detecting the presence of malware that uses DGA (domain generation algorithm) requires analysis of (plain-text) DNS payload. Therefore, solutions in this space typically train binary classifiers to detect specific attacks. For example, several ML algorithms for DDoS detection are evaluated considering both accuracy and inference latency in (Chi et al. 2024). Similarly, prior works have proposed methods for classifying DGA domains (Cebere et al. 2024), identifying bot traffic to servers using GAN-based data augmentation (Jan et al. 2020), and detecting encrypted malware communications over TLS (Anderson and McGrew 2017) as well as Tor (Dodia et al. 2022).

2.3 IoT device identification

Identifying IoT devices connected to a network is necessary for security auditing and for configuring appropriate policies (e.g., firewall rules) for effective management of the network. The challenge lies in determining the type of devices connected to the network through *traffic analysis*. In an enterprise network, a device identification solution would analyze north-south traffic as well as east-west traffic to identify devices that communicate both internally and externally. When labels for IoT devices and their corresponding network data are available, the task transforms into a multi-class classification problem.

AI models: IoT device identification has received considerable attention over the past decade, resulting in the proposal of various supervised and semi-supervised approaches for classifying known and unknown devices, respectively. They include several works that are based on conventional ML as well as DL models (Meidan et al. 2017; Nguyen et al. 2019b; Thangavelu et al. 2019; Dong et al. 2020; Wu et al. 2024). A recent work (Yu et al. 2020) demonstrates that devices connected to a WiFi network can be identified by training a DL model on the broadcast/multicast traffic observed.

Device identification is considered a defensive security measure, as organizations (Wu et al. 2024) and individuals (Sharma et al. 2022) need to track the devices connected to their networks. However, when viewed from the perspective of a home or public user, adversarial identification of consumer devices—whether using a compromised router or through a public WiFi hotspot—represents a violation of user privacy. This has led to the development of counter-fingerprinting techniques, e.g., using generative adversarial perturbations (Shenoi et al. 2023).

2.4 Website fingerprinting (WFP) attack

WFP attack targets user privacy by aiming to identify the websites a user visits. The research efforts in this space shed light on the vulnerability of existing and new network protocols in revealing sensitive user information. Since the focus is on web traffic, the relevant network protocols for traffic modeling are primarily limited to DNS (and its encrypted variants) and HTTPS. However, as users may employ VPNs or Tor (Dingledine et al. 2004) to mitigate such privacy attacks, research proposals typically assume that the traffic is tunneled through these services. The targets of WFP attacks can include anyone, such as citizens of interest, journalists, activists, or politicians, who may be monitored (or surveilled) by those in power. Consequently, the threat model assumes an adversary that has access to a network node, such as a router (either through direct control or compromise). Thus the adversary can passively monitor and analyze network traffic to identify the websites visited by the target users.

AI models: Given that a browsing session consists of hundreds or even thousands of packets, this topic has witnessed significant development of deep learning-based attack strategies. In other words, the ability of deep learning models to learn from a large feature space represents a substantial advantage for this task. The literature has interesting WFP attack proposals that use various DL architectures ranging from CNN to AE to sequence models such as LSTM and transformer (Rimmer et al. 2018; Oh, Sunkam, and Hopper 2019; Bhat et al. 2019; Cherubin, Jansen, and Troncoso 2022; Smith, Mittal, and Perrig 2021; Deng et al. 2023; Jin et al. 2023; Mitseva and Panchenko 2024; Csikor et al. 2025). There are also strategies proposed for countering them (Gong and Wang 2020; Smith et al. 2022; Gong et al. 2022; Siby et al. 2023; Shen et al. 2023). These models are evaluated under two scenarios: i) a *closed-world* (lab) set-

ting, where a classification model is trained to classify n specific *monitored* websites (each with samples collected over time) and is tested for accurate classification of the same n websites, where n is typically in a few hundreds; ii) a more realistic *open-world* setting, where the challenge is to identify n *monitored* websites under the assumption that the user visits m ($m \gg n$) *unmonitored* websites, where m is in tens of thousands. Since tunneling of traffic hides packet headers, (different from the previous tasks) the features used here are the basic (or *raw*) ones such as packet-size, inter-arrival time (IAT) (or time lag) of packet and direction of packet (as browsing generates bidirectional traffic).

2.5 Other tasks and models

There are a numerous other `NetS&P` tasks; due to space constraints, we provide a brief outline here.

- **Censorship and Anonymity.** Nation-state adversaries carry out traffic analysis for surveillance and censorship. The Tor anonymity network (Dingledine et al. 2004) is developed to offer anonymized access to Internet users. However, Tor is prone to de-anonymization attack (Nasr, Bahramali, and Houmansadr 2018; Oh et al. 2022). A most recent proposal (Wu et al. 2025) slices flows into windows and employs attention-based MIL (multi-instance learning) to learn the most relevant segments of potentiality noise traffic flows, so as to correlated flows at the entry and exit nodes of a Tor network.
- **Token inference attack.** In this attack, the encrypted traffic is passively monitored to estimate the token sizes from response packets, to subsequently use to to infer the response from an AI assistant (Weiss, Ayzenshteyn, and Mirsky 2024), using a fine-tuned T5 transformer model (Raffel et al. 2020) is fine-tuned.

We also highlight the application of adversarial ML, which aims to generate adversarial samples for evading systems utilizing AI models. In the context of `NetS&P`, an adversarial ML approach can be exploited not only to bypass AI-based security systems but also to counter threats related to privacy and censorship. For instance, authors in (Shenoi et al. 2023) adapt generative adversarial perturbations (Poursaeed et al. 2018) for evading IoT fingerprinting solutions that employ AI models. And in the space of website fingerprinting, a defense based on GAN, specifically, WGAN-div (Wu et al. 2018), to evade WFP attacker (discriminator) is studied in (Gong et al. 2022).

3 Challenges (and Opportunities) Ahead

3.1 Data challenges

One of the most important questions is, *do the current AI-based solutions generalize beyond the experiments conducted in controlled lab environments?* Addressing this question is challenging due to the lack of *high-quality labeled* data. Most research on network anomaly detection, attack classification, and device identification relies on openly available labeled datasets. However, since these datasets are generated in lab environments, they often contain artifacts or 'bad design smells' (Flood et al. 2024) that raise concerns on model over-fitting and biases. In (Flood et al. 2024), the authors analyze seven highly-cited datasets and provide evidence of such artifacts in network intrusion detection research. For instance, in one dataset with eight attack classes, two basic features exhibit minimal overlap between benign and attack traffic. As a result, a simple perturbation of these two features—unrelated to the inherent characteristics of attacks—can evade multiple ML and DL models. This simple approach outperforms a state-of-the-art adversarial attack (Sheatsley et al. 2022) employing evolutionary computation and generative adversarial networks. This finding suggests that evaluations in (Sheatsley et al. 2022) may not provide a "meaningful measure of the attack's effectiveness" (Flood et al. 2024) due to dataset flaws. Similar biases in network datasets have also been highlighted for other `NetS&P` tasks, e.g., website fingerprinting (Jansen and Wails 2023; Jansen, Wails, and Johnson 2024b,a).

The issue at hand is not about the existing proposals per se, but rather the quality of the datasets used for evaluating network security tasks. Going one step further, the fundamental challenge is the difficulty of validating the quality network traffic. The traffic generated by a single day's communications in an enterprise network can reach hundreds of gigabytes, and when considering this scale over multiple days, across various enterprises and different verticals, the volume becomes staggering. Therefore, collecting and labeling real-life network traffic flows is an obstacle. Furthermore, real-world network traffic contains sensitive user information, and sharing this data poses significant privacy risks and ethical concerns. Model-based attacks can extract hidden patterns as fingerprinting attacks in the previous section reveal; meta-data from packet headers (excluding payloads) is sufficient to infer sensitive information. Consequently, it is understandable that enterprises and telco operators refrain from publicly sharing traffic collected from their networks for research and development purposes.

> There is currently a lack of real-world quality datasets with labels for `NetS&P` tasks such as anomaly detection, attack detection, website fingerprinting, etc.

This leaves the research community with the second-best option—generating traffic datasets in controlled lab environments. There are promising directions here. The aforementioned in-depth analysis of open datasets represents an important step forward; it also sheds light on how to evaluate the quality and fidelity of lab-generated datasets. Learning of broad principles will guide in framing the right questions for specific `NetS&P` tasks. For example, if SSH brute-force attempts are present in the attack category, the benign category should also have normal SSH flows; otherwise encoding the port (22) would classify all SSH flows as malicious and (falsely) validate the model!

> We need to establish design principles for generating real traffic data in controlled environments, and the data generated should be evaluated for fidelity.

As second direction, we note several recent research

proposals put forth to *synthesize* traffic data with high fidelity (Jan et al. 2020; Yin et al. 2022; Huang et al. 2023; Gong et al. 2024; Jiang et al. 2024; Cüppers et al. 2024). In (Huang et al. 2023), the authors propose using a Variational Autoencoder (VAE) to learn the flow-size distribution from datacenter network traces, subsequently generating sequences of flow sizes with a Recurrent Neural Network (RNN) using Gated Recurrent Units (GRU). In (Gong et al. 2024), a transformer model is used to estimate missing data in a network telemetry time-series data by correlating information from multiple sources. Although they do not directly address security problems, we can take a leaf out of them to understand the key requirements that have to be met to synthesize data for `NetS&P` tasks:

Synthetic Data Generation:

- How critical is the leakage of distribution parameters in the context of enterprise network data?
- How can models learn with minimal linkage of distribution parameters to the source network?
- What impact does constrained data synthesis have on performance of `NetS&P` solutions?

Another promising direction is *emulating* applications that generate network traffic in a controlled environment. This is a common practice in website fingerprinting domain, where thousands of websites are visited (over Tor and VPN) using browsers, to generate the corresponding browsing traffic (e.g., HTTPS packets). However, for training models for securing a consumer or enterprise network, broader set of applications have to be emulated. A recent proposal explores the possibility of generating network traffic by orchestrating public GitHub repositories (Bühler et al. 2022). Building on this concept, the authors of (Khan et al. 2024) propose a data-generation platform that can be managed using SDN. We are still in early stages, and further research is needed to validate the traffic thus generated, and subsequently also to explore generating malware and anomalous traffic with high fidelity and diversity.

Emulation of benign and malicious traffic:

- How do we emulate applications corresponding to different enterprises (e.g., finance vs. academia) with varying user characteristics, for generating benign application data?
- How do we carefully and ethically emulate malware behavior, including infection, lateral movement, C&C communications, data exfiltration, and other activities, for the purpose of generating malicious data?
- How can SDN help to dynamically configure and manage networks for traffic generation? (Daneshamooz et al. 2025)

3.2 Practical deployment

Real-time per-packet inference in network traffic analysis faces several practical limitations. Enterprise networks operate at tens of Gbps whereas telco networks operate at 100s of Gpbs. Even at 10 Gbps, the time available for per-packet decision-making is constrained to less than 100 nanoseconds per packet, which poses challenges for complex computations required for DL-based solutions. In fact, an expensive step in building an ML/DL pipeline is parsing packets for high-speed feature extraction (Zhou et al. 2023).

A single packet typically does not provide sufficient information to make informed decisions about its legitimacy. For instance, a TCP SYN packet can be part of a legitimate connection or a malicious attempt.Only by *aggregating* packets, e.g., by source/destination IP address/port and segmented by time, can patterns indicative of attacks, such as TCP SYN floods, be identified. However, storing and processing these *packet aggregates* require substantial resources, a challenge that increases with the network rate.

Packet aggregates serve as meaningful units for modeling traffic. However, extracting information—such as total volume or duration—from packet aggregates (e.g., 5-tuple flows or sessions grouped by src/dst IP) still necessitates processing each individual packet, which becomes increasingly challenging as network rates rise.

In this context, programmable data planes offer promising solutions for in-network computation at rates of terabits per second (Tpbs) (Sapio et al. 2017; Zhang et al. 2020; Liu et al. 2021; Michel et al. 2021). These are essentially switches and smartNICs that allow programmability of packet processing pipelines directly on the network devices while constraining to specific packet processing behavior, e.g., in terms of latency and throughput. Indeed, there have been attempts to implement tree-based models in programmable switches as that aligns well with the match-action pipeline of the programmable switch architecture (Xiong and Zilberman 2019; Zhou et al. 2023; Parizotto et al. 2023; Zheng et al. 2024).

However, the cost of processing every single packet, even when decisions are based on packet aggregates, remains a significant challenge (Seufert et al. 2024). Sampling has long been a viable solution for various network tasks, such as heavy-hitter monitoring (Estan and Varghese 2003). Nonetheless, applying sampling to model network traffic for security tasks requires further investigation. Developing models that can effectively learn from and infer dynamic features is important. Additionally, we have to design intelligent sampling strategies tailored to security use cases. For instance, protocol handshakes (TCP, TLS, etc.) are carried out in the initial part of a connection, and therefore might reveal important information (Barradas et al. 2024). This raises a research question—would an *adapting* sampling strategy that samples different parts of a packet sequence at different probabilities be better than static sampling methods (Claise, Trammell, and Aitken 2013)? To answer this, we need to explore the trade-offs between sampling methods and their impact on model performance in detecting security threats. With programmable data planes, we now have the possibility to develop an in-network solution that makes both decisions—sampling and inference—without leaving the data plane.

- Programmable data planes hold promise for implementing in-network DL-based solutions.
- Intelligent sampling, combined with models that can effectively handle missing data points, need to explored for NetS&P tasks.

3.3 Explainability of predictions

Explainability technically allows us to interpret the behavior of models, or in other words, decisions made by models. The capability to explain the model predictions is important in cybersecurity to build trust between model developer, end user and stakeholders (Bhatt et al. 2020). Explainable models also aid in improving performance by identifying and mitigating potential pitfalls, such as biases, vulnerabilities to out-of-sample data, and overfitting (Jacobs et al. 2022; Flood et al. 2024).

Consider different cybersecurity use cases. For instance, in malware analysis, an explainable model highlights specific API calls that perform malicious actions or endpoint registers that were modified, providing actionable insights for mitigation (Saqib et al. 2024). In phishing detection, an explanation-based solution can reveal discrepancies between the brand displayed on a webpage and its domain name, helping users understand why a website is flagged as suspicious (Lee et al. 2024). For provenance graph modeling of endpoint events (Mukherjee et al. 2023), a SOC (Security Operations Center) analyst would find it helpful if a model identifies the sequence of processes and events that led to the alert raised, such as *executing a dropped binary that connects to a malicious IP*.

The above examples illustrate the utility of *local explanations* that explain (the features leading to) inference decision for the given input samples. On the other hand, with *global explanations*, the goal is to decipher the overall understanding of the model; e.g., knowing certain expensive features do not increase the detection capability is useful to reduce the cost of deployment (Chakraborty et al. 2021; Seufert et al. 2024). We refer readers to (Zhao et al. 2024) for a detailed breakdown of these two categories of explanations and understanding of the current state-of-the-art techniques for explainability. Some of these techniques have been recently explored in the network security community, concentrating mostly on anomaly and attack detection (Nguyen et al. 2019a; Jacobs et al. 2022; Wei et al. 2023; Han et al. 2024). The state-of-the-art DL solutions for NetS&P tasks, e.g., (Deng et al. 2023; Wu et al. 2024; Weiss, Ayzenshteyn, and Mirsky 2024; Wu et al. 2025), are primarily based on the transformer model architecture (Vaswani et al. 2017); besides, transformer-based foundation models are also emerging for NetS&P tasks (see below). This indicates that the existing techniques and challenges related to explainability in transformer models are pertinent to NetS&P.

We also have to take consider the perspective of the end-user to understand what needs to be 'explained' along with the inference results. And this is task dependent. In anomaly or attack detection, a SOC analyst needs to understand why a particular traffic session is deemed malicious. This requires going beyond ranking important features (used for input representation) (Nguyen et al. 2019a; Jacobs et al. 2022). Indeed the prediction should be accompanied with concrete explanations, such as noting that 'there is a series of connection attempts from this enterprise endpoint that failed,' indicating a C&C connection attempt from an infected endpoint. A potential approach might be to utilize LLMs to 'translate' input feature-level explanations to a format consumable by SOC analysts.

Explainability also provides insights into adversarial ML attack capability (and thereby defense); e.g., DGA classifiers are known to be prone to adversarial attacks (Cebere et al. 2024). In counter-censorship models, explaining the results would aid in understanding the censorship strategy used (Frolov, Wampler, and Wustrow 2020; Alice et al. 2020; Xue et al. 2024)—such as whether the censorship technique in use depends highly on the features of the first few packets (Alice et al. 2020)—so as to build effective models for evading censorship. Knowing which features contribute to the accuracy of a model (Iglesias and Zseby 2015; Chakraborty et al. 2021) is useful in estimating the cost of deploying a solution.

- Understanding task-specific user requirement is important to determine the explainability capability we expect from a model.
- User studies (Apruzzese, Laskov, and Schneider 2023) help to gain a better understanding of requirements from stakeholders on explainable models for specific use cases.

3.4 Many models; convergence on the horizon

For more than two decades, network traffic modeling progressed alongside the advancements in AI, resulting in the development of hundreds of models for NetS&P tasks. We moved from simple statistical methods to ML classifiers such as Decision Trees to more complex and powerful DL models such as transformers; this evolution has been instrumental in training models with bigger datasets and larger feature space, allowing researchers to also define numerous features for various tasks.

Given that network protocols are clearly defined, and substantial progress has been made over the past decades in developing AI-based solutions for NetS&P tasks, it can be argued that the *fundamental* features useful for modeling are well understood. For illustration, consider a set of research works across the aforementioned NetS&P tasks—anomaly detection (Mirsky et al. 2018; Nguyen et al. 2019a), flow classification for botnet detection (Barradas et al. 2021), muti-class attack classification on IoT devices (Sudheera et al. 2021), multi-tab website fingerprinting (Deng et al. 2023), flow-correlation attack on Tor networks (Oh et al. 2022), video fingerprinting (Schuster, Shmatikov, and Tromer 2017; Sabzi et al. 2024), dApp fingerprinting (Shen et al. 2021), prompt inference (Weiss, Ayzenshteyn, and Mirsky 2024) and Tor de-anonymization attack (Wu et al. 2025)—the *raw* or *fundamental* packet-level features at the finest granularity used are packet-size, inter-arrival time (IAT) between packets, direction, protocol, and representa-

tions for src/dst IP addresses and port numbers. Features at a coarser granularity, such as flow-level features, are derived from these raw features; for example, flow volume (duration) is the sum of sizes (IAT) of packets in that flow[1].

State-of-the-art:

- Fundamental features from raw encrypted packets remain largely consistent across various NetS&P tasks.
- DL models today are capable of handling huge datasets and large feature space.

A natural next challenge is to go beyond having multiple models for the different tasks and build a foundation model for network traffic analysis.

The concept of foundation models recently gained attention in the network community (Le et al. 2022; Lin et al. 2022; Peng et al. 2024; Divakaran and Peddinti 2025; Guthula et al. 2025). Once a foundation model has been built, it can be applied to different NetS&P tasks, say, by fine-tuning with task-specific data. We argue that, for a foundation network model to be truly effective, it must possess several key properties that enhance its practical utility:

- Different network environments—consumer, enterprise, datacenter, and telco (5G network)—have different capacities and characteristics, leading to distinct methods of traffic capture. Cost of extracting and storing traffic features is a primary constraint, influenced by the speed and scale of the network—1 Gbps consumer bandwidth with a few nodes compared to a telco network operating at hundreds of Gbps with millions of subscribers. Therefore, a flexible foundation model is necessary to make inferences on traffic captured at different scales represented at multiple levels: i) raw packet features, ii) flows or 5-tuple packet aggregates, and iii) sessions or flow aggregates (Wu, Divakaran, and Gurusamy 2025). This flexibility enables the model to be deployed across networks with diverse data rates and capabilities. For instance, a home consumer network might have the capability to extract raw features from all packets, thereby deriving flow and session features, and thus making all *views* available. In contrast, a telco network, handling much higher volumes of traffic, might only collect aggregate features, such as IP-based session-level data.
- Following the unsupervised pretraining process in NLP (Devlin et al. 2019) (e.g., using masked language modeling), foundation network model could be trained in a self-supervised way. However, fine-tuning to specific use cases requires labeled data. Labeling flows and packets is not only labor-intensive but also prone to errors. Therefore, a foundational model should be designed to handle missing labels during fine-tuning for specific tasks, enabling it to adapt effectively even when complete annotations are not available.
- As multiple foundation models are likely to emerge out of research, each potentially trained on different datasets—some openly available (e.g., MAWI (Cho, Mitsuya, and Kato 2000) and CAIDA (The CAIDA UCSD 2025) and others private (Lin et al. 2022; Peng et al. 2024)–it becomes important to explore methods for combining these models effectively for better cost-effective utilization. Simultaneously, we must address the challenge of minimizing the risk of sensitive information leakage when integrating these models, ensuring that the benefits of collaboration are realized while maintaining data privacy and security.
- Explainability is an active and challenging issue in foundation models today (Zhao et al. 2024). When designing a foundation model for NetS&P tasks, it is crucial to prioritize explainability. The challenge lies in utilizing the foundation model for various downstream tasks while also providing clear interpretations of its decisions.

Key properties of a foundation model for network traffic:

- Representations capturing different traffic characteristics at different scales and views.
- Capability to handle missing and noisy labels.
- Collaboration and openness, while ensuring user privacy is protected, for the effective utilization of multiple foundation models.
- Explainability for the downstream tasks.

4 Conclusion

In this work, we highlighted the advancements in traffic modeling for NetS&P tasks, with the aim of understanding and identifying the challenges that lie ahead. We hope this discussion serves as a catalyst for defining relevant research problems and generating new ideas, ultimately enhancing the likelihood of developing deployable and effective models for securing networks and users.

5 Acknowledgment

This research/project is supported by the National Research Foundation, Singapore, and the Cyber Security Agency of Singapore under the National Cybersecurity R&D Programme and the CyberSG R&D Programme Office (Award CRPO-GC2-ASTAR-001). Any opinions, findings, conclusions, or recommendations expressed in these materials are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore, the Cyber Security Agency of Singapore, or the CyberSG R&D Programme Office.

[1]To be sure, there are a few exceptions. For example, DGA detection is based on DNS payloads, which however would not be available when encrypted DNS protocols are employed. Another exception is TLS feature extraction for representing session-level information. However, TLS is ubiquitous, and the transition from TLS 1.2 to TLS 1.3 has made most features unavailable, thanks to the encryption of handshakes and introduction of ECH. The estimation of certificate size that might be useful for C&C detection (Barradas et al. 2024), could be learned by training on packet-sequences with size being one of the features.

References

Alice; Bob; Carol; Beznazwy, J.; and Houmansadr, A. 2020. How China Detects and Blocks Shadowsocks. In *Proc. IMC*.

Anderson, B.; and McGrew, D. 2017. Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. In *Proc. ACM KDD*.

Apruzzese, G.; Laskov, P.; and Schneider, J. 2023. SoK: Pragmatic assessment of machine learning for network intrusion detection. In *Proc. IEEE EuroS&P*, 592–614.

Barradas, D.; Novo, C.; Portela, B.; Romeiro, S.; and Santos, N. 2024. Extending C2 Traffic Detection Methodologies: From TLS 1.2 to TLS 1.3-enabled Malware. In *Proc. RAID*.

Barradas, D.; Santos, N.; Rodrigues, L.; Signorello, S.; Ramos, F. M.; and Madeira, A. 2021. FlowLens: Enabling Efficient Flow Classification for ML-based Network Security Applications. In *Proc. NDSS*.

Bhat, S.; Lu, D.; Kwon, A.; and Devadas, S. 2019. Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning. *Priv. Enhancing Technol.*, 2019(4): 292–310.

Bhatt, U.; Xiang, A.; Sharma, S.; Weller, A.; Taly, A.; Jia, Y.; Ghosh, J.; Puri, R.; Moura, J. M. F.; and Eckersley, P. 2020. Explainable machine learning in deployment. In *Proc. Conf. on Fairness, Accountability, and Transparency*.

Brauckhoff, D.; Salamatian, K.; and May, M. 2009. Applying PCA for traffic anomaly detection: Problems and solutions. In *IEEE INFOCOM*, 2866–2870.

Bühler, T.; Schmid, R.; Lutz, S.; and Vanbever, L. 2022. Generating representative, live network traffic out of millions of code repositories. In *Proc ACM HotNets*.

Cebere, B. C.; Flueren, J. L. B.; Sebastián, S.; Plohmann, D.; and Rossow, C. 2024. Down to earth! Guidelines for DGA-based Malware Detection. In *Proc. RAID*.

Chakraborty, B.; Divakaran, D. M.; Nevat, I.; Peters, G. W.; and Gurusamy, M. 2021. Cost-Aware Feature Selection for IoT Device Classification. *IEEE Internet of Things Journal*.

Cherubin, G.; Jansen, R.; and Troncoso, C. 2022. Online website fingerprinting: Evaluating website fingerprinting attacks on tor in the real world. In *Proc. USENIX Security Symposium*.

Chi, K.; Xie, X.; Hu, Y.; Zhao, D.; Xie, Y.; Zhang, L.; and Cui, Y. 2024. E-DDoS: An Evaluation System for DDoS Attack Detection. In *Proc. IEEE ICNP*.

Cho, K.; Mitsuya, K.; and Kato, A. 2000. Traffic Data Repository at the WIDE Project. In *Proc. USENIX ATC*.

Claise, B.; Trammell, B.; and Aitken, P. 2013. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. STD 77, RFC Editor.

Csikor, L.; Lian, Z.; Zhang, H.; Lakshmanan, N.; and Divakaran, D. M. 2025. DNS-over-QUIC and HTTP/3 in the Era of Transformers: The New Internet Privacy Battle. *IEEE Communications Magazine*.

Cüppers, J.; Schoen, A.; Blanc, G.; and Gimenez, P.-F. 2024. FlowChronicle: Synthetic Network Flow Generation through Pattern Set Mining. *Proc. ACM Netw.*, 2(CoNEXT4).

Daneshamooz, J.; Guthula, S.; Nguyen, J.; Chen, W.; Chandrasekaran, S.; Gupta, A.; Gupta, A.; and Willinger, W. 2025. NETREPLICA: Toward a Programmable Substrate for Last-Mile Data Generation. arXiv:2507.13476.

Deng, X.; Yin, Q.; Liu, Z.; Zhao, X.; Li, Q.; Xu, M.; Xu, K.; and Wu, J. 2023. Robust multi-tab website fingerprinting attacks in the wild. In *IEEE S&P*, 1005–1022.

Devlin, J.; Chang, M. W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. NAACL-HLT*.

Dingledine, R.; Mathewson, N.; Syverson, P. F.; et al. 2004. Tor: The second-generation onion router. In *Proc. USENIX security symposium*, volume 4, 303–320.

Divakaran, D. M.; Fok, K. W.; Nevat, I.; and Thing, V. L. 2017. Evidence gathering for network security and forensics. *Digital Investigation*. DFRWS.

Divakaran, D. M.; and Peddinti, S. T. 2025. Large Language Models for Cybersecurity: New Opportunities. *IEEE Security & Privacy*, 23(05): 38–45.

Dodia, P.; AlSabah, M.; Alrawi, O.; and Wang, T. 2022. Exposing the rat in the tunnel: Using traffic analysis for Tor-based malware detection. In *Proc. CCS*.

Dong, S.; Li, Z.; Tang, D.; Chen, J.; Sun, M.; and Zhang, K. 2020. Your smart home can't keep a secret: Towards automated fingerprinting of IoT traffic. In *Proc. ACM AisaCCS*.

Estan, C.; and Varghese, G. 2003. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Trans. on Computer Systems (TOCS)*, 21(3): 270–313.

Flood, R.; Engelen, G.; Aspinall, D.; and Desmet, L. 2024. Bad Design Smells in Benchmark NIDS Datasets. In *IEEE EuroS&P*.

Frolov, S.; Wampler, J.; and Wustrow, E. 2020. Detecting Probe-resistant Proxies. In *Proc. NDSS*.

Gong, F.; Raghunathan, D.; Gupta, A.; and Apostolaki, M. 2024. Zoom2Net: Constrained Network Telemetry Imputation. In *Proc. ACM SIGCOMM Conference*.

Gong, J.; and Wang, T. 2020. Zero-delay Lightweight Defenses against Website Fingerprinting. In *USENIX Security Symposium*.

Gong, J.; Zhang, W.; Zhang, C.; and Wang, T. 2022. Surakav: Generating Realistic Traces for a Strong Website Fingerprinting Defense. In *Proc. IEEE S&P*.

Google. 2025a. Email encryption in transit. In *Google Transparency Report*.

Google. 2025b. HTTPS encryption on the web. In *Google Transparency Report*.

Guthula, S.; Beltiukov, R.; Battula, N.; Guo, W.; Gupta, A.; and Monga, I. 2025. netFound: Foundation Model for Network Security. arXiv:2310.17025.

Han, D.; Wang, Z.; Feng, R.; Jin, M.; Chen, W.; Wang, K.; Wang, S.; Yang, J.; Shi, X.; Yin, X.; and Liu, Y. 2024. Rules Refine the Riddle: Global Explanation for Deep Learning-Based Anomaly Detection in Security Applications. In *Proc. ACM CCS*.

Huang, S.; Peng, L.; Wang, M.; Liu, Y.; Liu, Z.; Wang, X.; and Cui, Y. 2023. Datacenter Network Deserves Better Traffic Models. In *Proc. ACM HotNets*.

Iglesias, F.; and Zseby, T. 2015. Analysis of network traffic features for anomaly detection. *Machine Learning*, 59–84.

Jacobs, A. S.; Beltiukov, R.; Willinger, W.; Ferreira, R. A.; Gupta, A.; and Granville, L. Z. 2022. AI/ML for Network Security: The Emperor has no Clothes. In *Proc. ACM CCS*.

Jan, S. T.; Hao, Q.; Hu, T.; Pu, J.; Oswal, S.; Wang, G.; and Viswanath, B. 2020. Throwing Darts in the Dark? Detecting Bots with Limited Data using Neural Data Augmentation. In *Proc. IEEE S&P*, 1190–1206.

Jansen, R.; and Wails, R. 2023. Data-explainable website fingerprinting with network simulation. *Proc. PETS*.

Jansen, R.; Wails, R.; and Johnson, A. 2024a. A Measurement of Genuine Tor Traces for Realistic Website Fingerprinting. *arXiv preprint arXiv:2404.07892*.

Jansen, R.; Wails, R.; and Johnson, A. 2024b. Repositioning Real-World Website Fingerprinting on Tor. In *Proc. ACM Workshop on Privacy in the Electronic Society*.

Jiang, X.; Liu, S.; Gember-Jacobson, A.; Bhagoji, A. N.; Schmitt, P.; Bronzino, F.; and Feamster, N. 2024. Netdiffusion: Network data augmentation through protocol-constrained traffic generation. *Proc. of the ACM on Measurement and Analysis of Computing Systems*, 8(1): 1–32.

Jin, Z.; Lu, T.; Luo, S.; and Shang, J. 2023. Transformer-based Model for Multi-tab Website Fingerprinting Attack. In *Proc. ACM CCS*.

Khan, P. I.; Guthula, S.; Beltiukov, R.; Schmid, R.; Bühler, T.; Gupta, A.; Vanbever, L.; and Willinger, W. 2024. Harnessing Public Code Repositories to Develop Production-Ready ML Artifacts for Networking. In *Proc. Applied Networking Research Workshop*, 100–102.

Lakhina, A.; Crovella, M.; and Diot, C. 2004. Diagnosing Network-Wide Traffic Anomalies. *ACM SIGCOMM Comput. Commun. Rev.*, 34(4): 219–230.

Le, F.; Srivatsa, M.; Ganti, R.; and Sekar, V. 2022. Rethinking data-driven networking with foundation models: challenges and opportunities. In *Proc ACM HotNets*.

Lee, J.; Lim, P.; Hooi, B.; and Divakaran, D. M. 2024. Multimodal Large Language Models for Phishing Webpage Detection and Identification. In *Proc. Symposium on Electronic Crime Research (eCrime)*.

Lin, X.; Xiong, G.; Gou, G.; Li, Z.; Shi, J.; and Yu, J. 2022. ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification. In *Proc. ACM Web Conference*.

Liu, Z.; Namkung, H.; Nikolaidis, G.; Lee, J.; Kim, C.; Jin, X.; Braverman, V.; Yu, M.; and Sekar, V. 2021. Jaqen: A High-Performance Switch-Native approach for detecting and mitigating volumetric DDoS attacks with programmable switches. In *Proc. USENIX Security Symposium*.

Lyu, M.; Gharakheili, H. H.; and Sivaraman, V. 2022. A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques. *ACM Comput. Surv.*, 55(8).

Meidan, Y.; Bohadana, M.; Shabtai, A.; Guarnizo, J. D.; Ochoa, M.; Tippenhauer, N. O.; and Elovici, Y. 2017. ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis. In *Proc. of the Symposium on Applied Computing*, 506–509.

Michel, O.; Bifulco, R.; Rétvári, G.; and Schmid, S. 2021. The Programmable Data Plane: Abstractions, Architectures, Algorithms, and Applications. *ACM Comput. Surv.*, (4).

Mirsky, Y.; Doitshman, T.; Elovici, Y.; and Shabtai, A. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. In *Proc. NDSS*.

Mitseva, A.; and Panchenko, A. 2024. Stop, don't click here anymore: boosting website fingerprinting by considering sets of subpages. In *Proc. USENIX Security Symposium*.

Mukherjee, K.; Wiedemeier, J.; Wang, T.; Wei, J.; Chen, F.; Kim, M.; Kantarcioglu, M.; and Jee, K. 2023. Evading Provenance-Based ML Detectors with Adversarial System Actions. In *Proc. USENIX Security Symposium*.

Nasr, M.; Bahramali, A.; and Houmansadr, A. 2018. DeepCorr: Strong flow correlation attacks on tor using deep learning. In *Proc. ACM CCS*, 1962–1976.

Nevat, I.; Divakaran, D. M.; Nagarajan, S. G.; Zhang, P.; Su, L.; Ko, L. L.; and Thing, V. L. L. 2018. Anomaly Detection and Attribution in Networks With Temporally Correlated Traffic. *IEEE/ACM Trans. Netw.*, 26(1): 131–144.

Nguyen, Q. P.; Lim, K. W.; Divakaran, D. M.; Low, K. H.; and Chan, M. C. 2019a. GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection. In *Proc. IEEE CNS*.

Nguyen, T. D.; Marchal, S.; Miettinen, M.; Fereidooni, H.; Asokan, N.; and Sadeghi, A.-R. 2019b. DÏoT: A federated self-learning anomaly detection system for IoT. In *Proc. ICDCS*.

Oh, S. E.; Sunkam, S.; and Hopper, N. 2019. p-FP: Extraction, classification, and prediction of website fingerprints with deep learning. *Proc. PETS*.

Oh, S. E.; Yang, T.; Mathews, N.; Holland, J. K.; Rahman, M. S.; Hopper, N.; and Wright, M. 2022. DeepCoFFEA: Improved flow correlation attacks on Tor via metric learning and amplification. In *Proc. IEEE S&P*.

Ozery, Y.; Nadler, A.; and Shabtai, A. 2024. Information based heavy hitters for real-time DNS data exfiltration detection. In *Proc. NDSS*, 1–15.

Parizotto, R.; Coelho, B. L.; Nunes, D. C.; Haque, I.; and Schaeffer-Filho, A. 2023. Offloading machine learning to programmable data planes: A systematic survey. *ACM Computing Surveys*, 56(1): 1–34.

Peng, L.; Xie, X.; Huang, S.; Wang, Z.; and Cui, Y. 2024. Ptu: Pre-Trained Model for Network Traffic Understanding. In *Proc. ICNP*.

Poursaeed, O.; Katsman, I.; Gao, B.; and Belongie, S. 2018. Generative adversarial perturbations. In *Proc. CVPR*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683.

Rimmer, V.; Preuveneers, D.; Juarez, M.; van Goethem, T.; and Joosen, W. 2018. Automated Website Fingerprinting through Deep Learning. In *Proc. NDSS*.

Sabzi, A.; Vora, R.; Goswami, S.; Seltzer, M.; Lécuyer, M.; and Mehta, A. 2024. NetShaper: A Differentially Private Network Side-Channel Mitigation System. In *Proc. USENIX Security Symposium*.

Sapio, A.; Abdelaziz, I.; Aldilaijan, A.; Canini, M.; and Kalnis, P. 2017. In-Network Computation is a Dumb Idea Whose Time Has Come. In *Proc. HotNets*.

Saqib, M.; Mahdavifar, S.; Fung, B. C.; and Charland, P. 2024. A Comprehensive Analysis of Explainable AI for Malware Hunting. *ACM Computing Surveys*, 56(12): 1–40.

Schuster, R.; Shmatikov, V.; and Tromer, E. 2017. Beauty and the burst: Remote identification of encrypted video streams. In *Proc. USENIX Security Symposium*.

Seufert, M.; Dietz, K.; Wehner, N.; Geißler, S.; Schüler, J.; Wolz, M.; Hotho, A.; Casas, P.; Hoßfeld, T.; and Feldmann, A. 2024. Marina: Realizing ML-Driven Real-Time Network Traffic Monitoring at Terabit Scale. *IEEE Trans. on Network and Service Management*, 21(3): 2773–2790.

Sharma, R. A.; Soltanaghaei, E.; Rowe, A.; and Sekar, V. 2022. Lumos: Identifying and Localizing Diverse Hidden IoT Devices in an Unfamiliar Environment. In *31st USENIX Security Symposium*.

Sheatsley, R.; Papernot, N.; Weisman, M. J.; Verma, G.; and McDaniel, P. 2022. Adversarial examples for network intrusion detection systems. *J. Comput. Secur.*, 30(5): 727–752.

Shen, M.; Ji, K.; Gao, Z.; Li, Q.; Zhu, L.; and Xu, K. 2023. Subverting website fingerprinting defenses with robust traffic representation. In *Proc. USENIX Security Symposium*.

Shen, M.; Zhang, J.; Zhu, L.; Xu, K.; and Du, X. 2021. Accurate decentralized application identification via encrypted traffic analysis using graph neural networks. *IEEE Trans. on Information Forensics and Security*.

Shenoi, A.; Vairam, P. K.; Sabharwal, K.; Li, J.; and Divakaran, D. M. 2023. iPET: Privacy Enhancing Traffic Perturbations for Secure IoT Communications. In *Proc. PETS*.

Siby, S.; Barman, L.; Wood, C.; Fayed, M.; Sullivan, N.; and Troncoso, C. 2023. Evaluating practical QUIC website fingerprinting defenses for the masses. In *Proc. PETS*.

Smith, J.; Mittal, P.; and Perrig, A. 2021. Website Fingerprinting in the Age of QUIC. *Proc. Priv. Enhancing Technol.*, 2021(2): 48–69.

Smith, J.-P.; Dolfi, L.; Mittal, P.; and Perrig, A. 2022. QCSD: A QUIC Client-Side Website-Fingerprinting Defence Framework. In *USENIX Security Symposium*.

Sudheera, K. L. K.; Divakaran, D. M.; Singh, R. P.; and Gurusamy, M. 2021. ADEPT: Detection and Identification of Correlated Attack Stages in IoT Networks. *IEEE Internet Things Journal*.

Thangavelu, V.; Divakaran, D. M.; Sairam, R.; Bhunia, S. S.; and Gurusamy, M. 2019. DEFT: A Distributed IoT Fingerprinting Technique. *IEEE Internet of Things Journal*.

The CAIDA UCSD. 2025. The CAIDA Anonymized Internet Traces Dataset (April 2008 - January 2019).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. In *Proc. NIPS*.

Wei, F.; Li, H.; Zhao, Z.; and Hu, H. 2023. xNIDS: Explaining Deep Learning-based Network Intrusion Detection Systems for Active Intrusion Responses. In *Proc. USENIX Security Symposium*.

Weiss, R.; Ayzenshteyn, D.; and Mirsky, Y. 2024. What Was Your Prompt? A Remote Keylogging Attack on AI Assistants. In *Proc. USENIX Security Symposium*.

Wu, B.; Divakaran, D.; Csikor, L.; and Gurusamy, M. 2025. RECTor: Robust and Efficient Correlation Attack on Tor. *IEEE Communications Magazine*.

Wu, B.; Divakaran, D. M.; and Gurusamy, M. 2025. UniNet: A Unified Multi-Granular Traffic Modeling Framework for Network Security. *IEEE Trans. on Cognitive Communications and Networking*.

Wu, B.; Gysel, P.; Divakaran, D. M.; and Gurusamy, M. 2024. ZEST: Attention-based Zero-Shot Learning for Unseen IoT Device Classification. In *IEEE NOMS*, 1–9.

Wu, J.; Huang, Z.; Thoma, J.; Acharya, D.; and Van Gool, L. 2018. Wasserstein divergence for GANs. In *Proc. ECCV*.

Xiong, Z.; and Zilberman, N. 2019. Do switches dream of machine learning? Toward in-network classification. In *Proc. ACM HotNets*, 25–33.

Xue, D.; Ramesh, R.; Jain, A.; Kallitsis, M.; Halderman, J. A.; Crandall, J. R.; and Ensafi, R. 2024. OpenVPN is Open to VPN Fingerprinting. *Commun. ACM*, 79–87.

Yin, Y.; Lin, Z.; Jin, M.; Fanti, G.; and Sekar, V. 2022. Practical GAN-based synthetic ip header trace generation using netshare. In *Proc. ACM SIGCOMM*.

Yu, L.; Luo, B.; Ma, J.; Zhou, Z.; and Liu, Q. 2020. You are what you broadcast: Identification of mobile and IoT devices from (public) WiFi. In *Proc. USENIX Security Symposium*.

Zenati, H.; Romain, M.; Foo, C.-S.; Lecouat, B.; and Chandrasekhar, V. 2018. Adversarially Learned Anomaly Detection. In *IEEE ICDM*.

Zhang, M.; Li, G.; Wang, S.; Liu, C.; Chen, A.; Hu, H.; Gu, G.; Li, Q.; Xu, M.; and Wu, J. 2020. Poseidon: Mitigating volumetric DDoS attacks with programmable switches. In *Proc. NDSS*.

Zhao, H.; Chen, H.; Yang, F.; Liu, N.; Deng, H.; Cai, H.; Wang, S.; Yin, D.; and Du, M. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2): 1–38.

Zheng, C.; Xiong, Z.; Bui, T. T.; Kaupmees, S.; Bensoussane, R.; Bernabeu, A.; Vargaftik, S.; Ben-Itzhak, Y.; and Zilberman, N. 2024. IIsy: Hybrid In-Network Classification Using Programmable Switches. *IEEE/ACM Transactions on Networking*, 32(3): 2555–2570.

Zhou, C.; and Paffenroth, R. C. 2017. Anomaly Detection with Robust Deep Autoencoders. In *Proc. ACM KDD*.

Zhou, G.; Liu, Z.; Fu, C.; Li, Q.; and Xu, K. 2023. An Efficient Design of Intelligent Network Data Plane. In *32nd USENIX Security Symposium*, 6203–6220.