Bridging High-Level Intent and Network Execution: Detecting Violations and Intent Drift Through Low-Level Traffic Analysis

Tonia Haikal*, Shereen Ismail†, Eman Hammad*,
* iSTAR Lab at Texas A&M University, College Station, TX 77840, USA.
† Merit Network, Inc., University of Michigan, Ann Arbor, MI 48108, USA.

***Abstract*—Intent-Based Networking (IBN) aims to bridge abstract administrative goals and autonomous network configurations. We formalize flow-level identifiers: IP addresses, MAC addresses, and ports, as Internal Low-Level Intents (ILI), the executable units of translated intent. Using large-scale telemetry data collected from a distributed honeynet system, we stress-test intent-to-flow translation in a high-entropy, adversarial environment. We evaluate three policy regimes: Strict, Balanced, and Permissive, using two metrics: policy violations and intent drift.**

Our analysis reveals that reductions in policy violations do not necessarily correspond to improved alignment between intended and observed network behavior, indicating that traditional violation metrics may reflect administrative permissiveness rather than operational stability. Furthermore, we observe non-uniform concentration of violations across service classes, suggesting that flow-level analysis can support granular, service-aware micro-segmentation. Ultimately, this work establishes low-level flow tuples as a measurable bridge for verifiable intent assurance in autonomous systems.

***Index Terms*—Intent-Based Networking, Intent Assurance, Network Security, Intent Drift, Policy Enforcement, Empirical Traffic Analysis.**

I. Introduction

Intent-Based Networking (IBN) shifts network management from manual configuration to goal-oriented autonomy. Its core objective is to allow administrators to express what the network should achieve while the system determines how to implement and enforce those requirements. Despite progress in intent translation, a significant gap remains in intent assurance: current systems can verify that a configuration was deployed, but they lack empirical frameworks to verify that resulting data-plane traffic adheres to the original human intent.

In practice, high-level intent is executed through low-level flow identifiers, including IP addresses, MAC addresses, and transport-layer ports. This paper leverages these identifiers as Internal Low-Level Intents (ILI), i.e., the executable representation of intent. To achieve reliable assurance, an autonomous system must bridge abstract policy declarations with high-entropy live network traffic.

In this work, we propose an empirical approach that uses large-scale telemetry collected from the Merit Network HoneyTrap-based honeynet system [1] to evaluate the relationship between intent and execution across three policy tiers: Strict, Balanced, and Permissive. Through this analysis, we introduce two metrics: Policy Violations and Intent Drift. This framework shows how low-level flow analysis can reveal divergence between policy compliance and operational behavior, supporting data-driven policy refinement in autonomous network systems.

II. Related Work and Background

Intent-Based Networking (IBN) enables operators to express high-level objectives while the system translates them into enforceable configurations. The IBN lifecycle is commonly described through intent translation, conflict resolution, assurance, and closed-loop management [2], [3]. Recent work has advanced IBN automation through policy and configuration platforms [4] and large language model assisted intent management [5]. However, much of this work focuses on translating intent rather than empirically validating whether resulting flow-level behavior remains consistent with the intended policy.

Security remains a central concern in intent-driven and programmable networks because automated enforcement can introduce new attack surfaces and policy ambiguities [6]. Existing datasets and frameworks often treat IP addresses, MAC addresses, ports, and protocols as configuration artifacts or traffic features rather than

measurable objects of translated intent [7]. In contrast, this work formalizes these identifiers as *Internal Low-Level Intents* (ILI) and uses them as the basis for data-plane assurance.

Empirical adversarial telemetry provides an important foundation for evaluating network behavior under realistic threat conditions. Honeypots and network telescopes have been used to observe unsolicited traffic, scanning, brute-force attempts, and service-targeting behavior at scale [8], [9]. Recent HoneyTrap-based studies further demonstrate the value of large-scale honeynet telemetry for characterizing adversarial activity and exposed-service behavior [1]. Building on this work, we use the HoneyTrap corpus to compare policy regimes, quantify violations, measure drift, and examine how service-level concentration can inform micro-segmentation decisions.

III. Proposed Intent-assurance Approach

Figure 1 illustrates the overall workflow of the proposed intent-assurance framework, beginning with large-scale HoneyTrap-based honeynet telemetry collection and progressing through data preprocessing, flow construction, policy evaluation, drift measurement, and security assurance generation. First, raw honeynet telemetry containing unsolicited and adversarial traffic is collected from distributed sensors and normalized into structured flow records. The framework then extracts ILI tuples contains source and destination IP addresses, MAC addresses, transport-layer ports, and protocol information. These flow tuples serve as the operational bridge between high-level network intent and observable data-plane behavior. Next, the observed flows are evaluated against multiple intent-policy tiers: Strict, Balanced, and Permissive policies, to identify policy violations based on predefined service constraints and allowed communication patterns. In parallel, the framework computes intent drift by comparing observed traffic flows against an empirical baseline of dominant recurring communication patterns. This separation enables the framework to distinguish between explicit policy non-compliance and behavioral deviations that may still occur under allowed policies. Finally, the resulting violation statistics, drift measurements, and service-level traffic concentrations provide actionable assurance insights that support policy refinement, service-aware micro-segmentation, and autonomous security enforcement in future intent-driven and 6G network environments. The following subsections provide further details regarding the proposed approach, implementation methodology, and experimental results.

A. From High-Level Intent to Executable Network State

Intent-Based Networking (IBN) enables administrators to specify desired network outcomes while the infrastructure translates those objectives into enforceable behavior. Although intent may be expressed as an abstract policy statement, enforcement ultimately occurs through concrete data-plane identifiers such as source and destination IP addresses, MAC addresses, transport-layer ports, and protocol information. In this work, we treat these identifiers as the operational realization of translated intent and model Internal Low-Level Intents (ILI) as flow-level tuples:

$$f = (srcIP, dstIP, srcMAC, dstMAC, srcPort, dstPort, protocol)$$

These tuples represent the executable layer through which high-level intent can be evaluated against observed data-plane behavior. This view is especially relevant in programmable and next-generation networks, where intent must eventually become actionable rules for switches, routers, firewalls, virtual network functions, or other policy enforcement points. Therefore, the traffic attributes in the dataset provide a measurable surface for evaluating whether observed behavior is consistent with translated intent.

B. Intent Translation as Structured Flow Representation

A central requirement of IBN is transforming human-readable policy statements into structured and enforceable configurations. Regardless of whether translation is performed through policy compilers, orchestration frameworks, or large language model assisted pipelines, the output must ultimately be expressed through network-level entities, service constraints, and communication paths.

In this work, translated intent is modeled as structured flow tuples defined over source and destination addresses, MAC identities, transport-layer ports, and protocol information. For each complete-flow record, we construct a flow key using the source IP, destination IP, destination port, and protocol. This key allows observed traffic to be compared against policy constraints and an empirical expected-flow baseline.

This representation supports two assurance properties: identifying flows that violate the selected policy tier and detecting flows that deviate from dominant recurring communication patterns. In this way, the framework moves from abstract intent translation toward measurable intent assurance using flow-level telemetry.

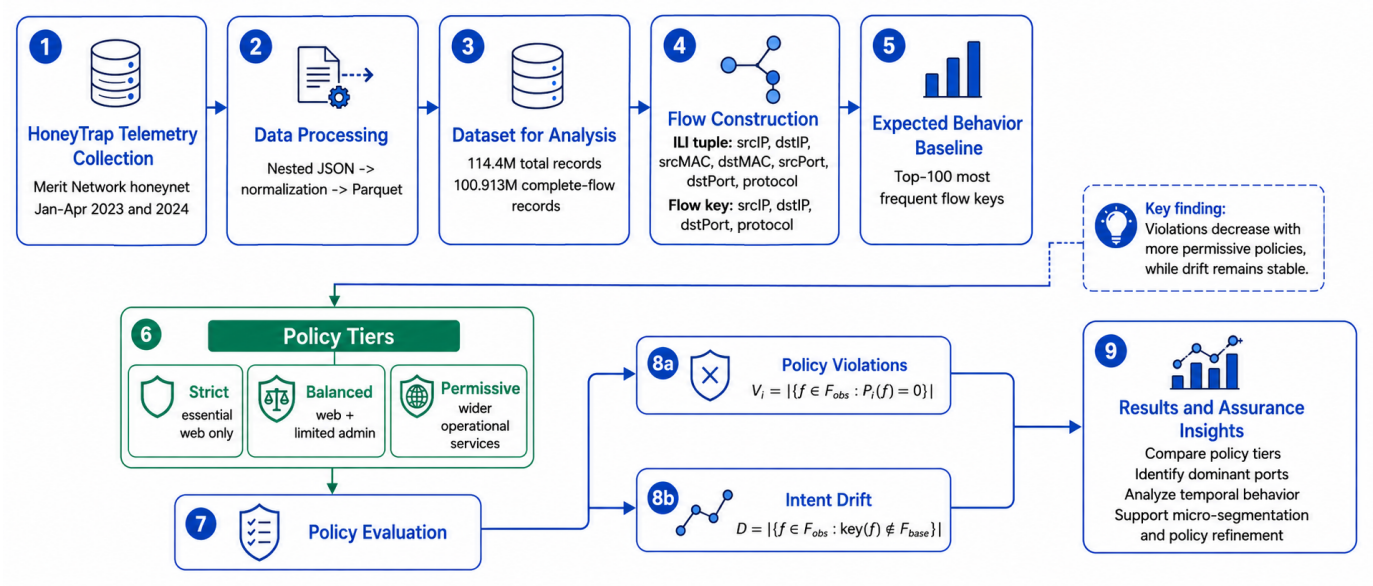

Fig. 1. Workflow of the proposed intent-assurance framework.

C. Dataset Description and Flow Construction

The raw HoneyTrap-based honeynet data is organized as nested JSON files by year, month, day, and hour. To support scalable analysis, these records are parsed, normalized, and consolidated into Parquet format. Table I summarizes the dataset used in this study, including the total number of records and the subset of complete-flow records used for intent-assurance analysis.

TABLE I
DATASET SUMMARY FOR INTENT-ASSURANCE ANALYSIS.

Dataset Attribute	Value
Raw data format	Nested JSON files
Organization	Year/Month/Day/Hour
Processed format	Parquet
Total records	114.4 million
Complete-flow records	100,913,000
Flow construction fields	srcIP, dstIP, dstPort, protocol, timestamp
Flow key fields	srcIP, dstIP, dstPort, protocol

For each complete-flow record, we construct a flow key using the source IP, destination IP, destination port, and protocol. This flow key is used to evaluate policy violations and intent drift in the subsequent analysis. To provide temporal context for the dataset, Figure 2 shows the distribution of activity across the first four months of 2023 and 2024. Activity is concentrated in the early months of each year, with the strongest intensity in January and lower intensity in March and April, which is consistent with the monthly violation and drift trends observed later in the evaluation.

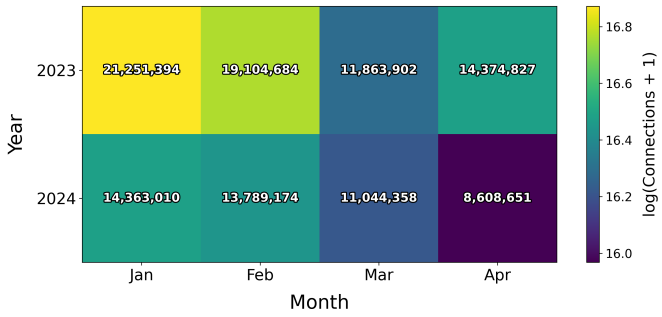

Fig. 2. Year-by-month activity heatmap for January through April.

D. Formal Definitions of Violation and Drift

To make the proposed assurance framework measurable, we define two metrics: policy violation and intent drift. Let F_{obs} denote the set of observed complete-flow records used for policy analysis. Each flow $f \in F_{obs}$ is represented as:

$$f = (srcIP, dstIP, srcMAC, dstMAC, srcPort, dstPort, protocol)$$

Let $P_i(f)$ denote the policy predicate for policy tier i, where $i \in \{\text{Strict}, \text{Balanced}, \text{Permissive}\}$. The predicate returns 1 if flow f is allowed under policy tier i and 0 otherwise. In this implementation, $P_i(f)$ is evaluated

using destination-port membership in the allowed and restricted service sets defined for each policy tier.

A policy violation occurs when an observed flow does not satisfy the active policy predicate. The total number of violations under policy tier i is defined as:

$$V_i = |\{f \in F_{obs} : P_i(f) = 0\}|$$

This metric captures explicit non-compliance with the selected enforcement policy and is expected to change as the allowed service set becomes more or less permissive.

Intent drift is defined using an empirical expected-flow baseline. For each observed flow, the implementation constructs a flow key from the source IP, destination IP, destination port, and protocol. Let K denote the set of observed flow keys. The baseline F_{base} is defined as the most frequent k flow keys in the complete-flow data set, with $k = 100$:

$$F_{base} = \text{Top}_{k=100}(K)$$

A flow is considered drifted when its flow key does not belong to this baseline. Total drift is therefore defined as:

$$D = |\{f \in F_{obs} : key(f) \notin F_{base}\}|$$

Unlike policy violations, which depend on the selected policy tier, F_{base} remains fixed across all policies. This distinction allows the evaluation to separate policy compliance from flow-behavior stability: violations measure whether traffic satisfies the active enforcement rules, while drift measures whether traffic deviates from the dominant recurring flow patterns.

E. Conflict Detection and Assurance of Translated Intent

When multiple intents coexist, translation ambiguity and policy overlap can introduce contradictions. For example, one intent may authorize access to a service while another restricts the same destination, port, or service class. Representing intent through low-level identifiers makes these conflicts observable as overlaps among addresses, ports, protocols, and service groups.

After deployment, observed complete-flow records are evaluated against policy predicates to identify violations and against a fixed empirical baseline to identify drift. This separation is important because a flow may comply with a permissive policy while still deviating from expected communication patterns. Thus, ILI-based assurance captures both enforcement compliance and behavioral stability.

F. Data-Driven Policy Construction

The dataset used in this analysis was collected from a HoneyTrap-based honeynet system deployed within the Merit Network infrastructure [1]. The telemetry captures unsolicited network activity observed by distributed sensors, including connection attempts, service interactions, destination ports, protocol information, and timestamped flow attributes. For temporal consistency, this study focuses on traffic collected during the first four months of 2023 and 2024, allowing policy behavior to be compared across matching seasonal windows.

The empirical analysis shows that violations cluster around a small subset of destination ports rather than being uniformly distributed across services. Port 25565 is the dominant outlier, while ports such as 5900, 179, 23, 53, 445, 30120, 21, 22, and 3389 also appear prominently, as shown in Figure 3. This concentration supports service-oriented policy groupings instead of a single monolithic enforcement rule.

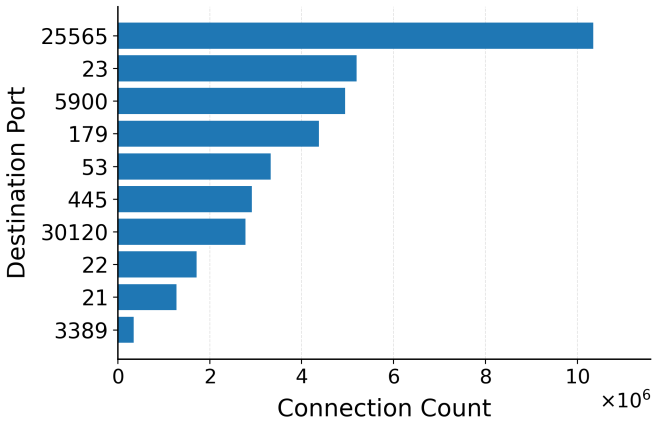

Fig. 3. Top Destination Ports by Connection Count

Based on these observations, three policy tiers were constructed to evaluate enforcement strictness. The *Strict Policy* allows only essential web services, the *Balanced Policy* allows standard web services and limited administrative access, and the *Permissive Policy* allows a wider set of operational services while preserving restrictions on sensitive ports.

G. Policy Violations Across Intent Policies

Figure 4 shows that total policy violations decrease as the allowed service set expands. The Strict Policy produces 95,024,343 violations, followed by the Balanced Policy with 92,988,515 and the Permissive Policy with 87,701,038. This monotonic decrease shows

that the framework responds consistently to enforcement strictness.

Figure 4 compares total violations against total drift. While violations decline across the three tiers, drift remains fixed at 89,031,223 flows because it is computed against a policy-invariant empirical baseline. This shows that policy compliance and flow-behavior stability are distinct assurance properties.

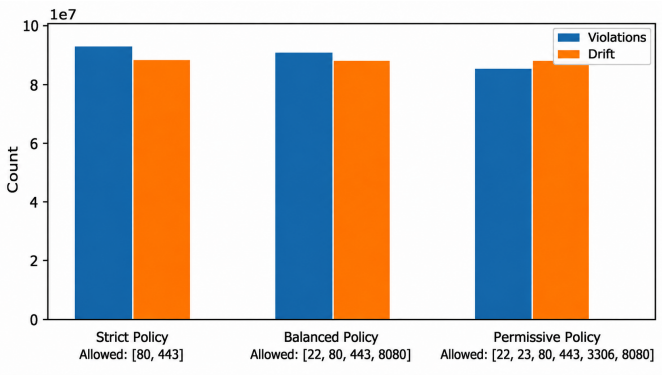

Fig. 4. Intent Policy Comparison: Total Violations and Intent Drift

H. Violation Composition and Dominant Services

Figure 5 shows that most violations fall into the non-allowlisted-port category across all policy tiers. Under the Strict Policy, this category accounts for 84,333,503 violations, while the Balanced and Permissive policies record 79,549,689 and 79,125,864 violations, respectively. This decreasing pattern is consistent with the policy definitions, since expanding the allowed service set reduces the number of flows classified as non-allowlisted. The joint non-allowlisted and restricted-service category follows the same trend, decreasing from 15,898,479 under the Strict Policy to 8,655,012 under the Balanced Policy and 8,151,349 under the Permissive Policy.

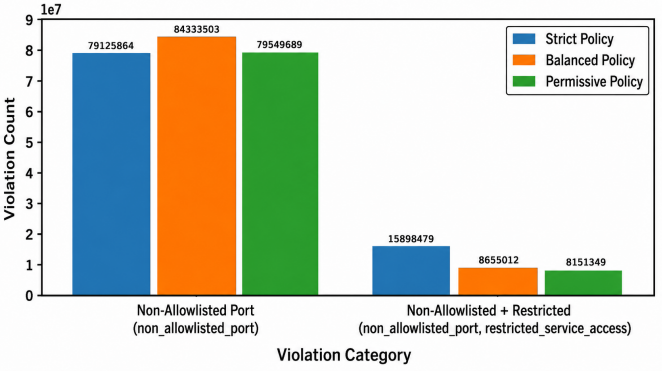

Fig. 5. Violation Type Distribution Across Intent Policies

Figure 6 shows the dominant destination ports associated with policy violations after applying the policy-specific allowlist rules. Port 25565 remains the largest violating destination port across all three policy tiers, indicating persistent high-volume traffic outside the intended service set. Other consistently violating ports include 5900, 179, 53, 445, 30120, 21, and 3389. Ports 22 and 23 vary by policy tier because they are permitted under more permissive configurations; therefore, they do not appear as violations for policies in which they are allowlisted. This result supports the use of low-level flow tuples for service-aware micro-segmentation and intent-assurance validation.

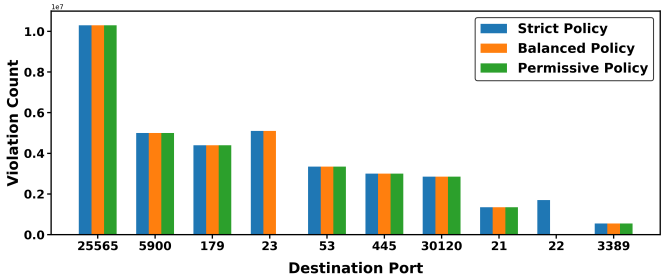

Fig. 6. Top Violating Destination Ports Across Intent Policies

I. Temporal Behavior of Violations and Drift

Figure 7 shows persistent month-to-month variation in policy violations, with peaks in January and March of both years and lower activity in April. The policy ranking remains stable over time: Strict is highest, Balanced is in the middle, and Permissive is lowest. This consistency supports the interpretation that the policy tiers capture meaningful enforcement differences.

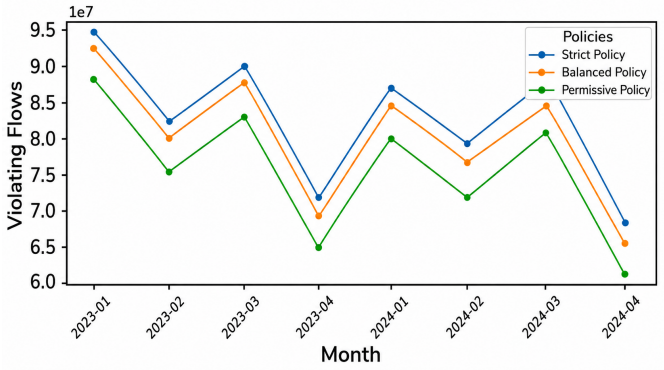

Fig. 7. Monthly Policy Violations Across Intent Policies

Figure 8 shows similar temporal fluctuation for drift. Although drift varies across months, it is evaluated independently of the selected policy tier in the current

framework. This supports the finding that reducing policy violations does not necessarily eliminate operational deviation.

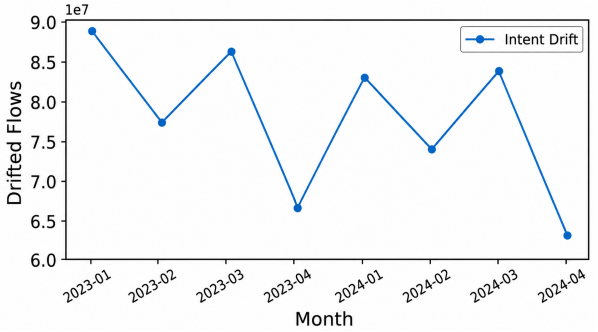

Fig. 8. Monthly Intent Drift Across Intent Policies

J. Security Implications for Intent-Based Networking

These results have direct implications for security in intent-driven environments. Because translated intent is represented through explicit flow-level identifiers, verification mechanisms can assess whether observed communication paths remain consistent with authorized policy. This supports role-based enforcement, detection of unauthorized service exposure, and identification of drifted communication paths.

The concentration of violations around a small number of service ports also supports micro-segmentation. Rather than treating intent assurance as a coarse allow/deny problem, the network can enforce service-aware controls across hosts, device roles, and trust zones. This is especially relevant to distributed 6G-style infrastructures, where low-level flow tuples provide a measurable boundary between abstract intent and operational network state.

K. Summary of the Approach

Overall, the proposed approach evaluates translated intent as an executable flow-level representation against large-scale observed traffic. Policy violations decrease from 95,024,343 under the Strict Policy to 87,701,038 under the Permissive Policy, while intent drift remains fixed at 89,031,223 flows across all tiers, as shown in Figure 4. This demonstrates that policy compliance and flow-behavior stability are distinct assurance properties, and that low-level flow tuples can support policy comparison and micro-segmentation analysis.

IV. Conclusion and Future Work

This paper presented a flow-level assurance framework for Intent-Based Networking (IBN) by modeling low-level network identifiers as Internal Low-Level Intents (ILI). Using large-scale HoneyTrap telemetry, we evaluated translated intent through three policy tiers: Strict, Balanced, and Permissive. The results show that policy violations decrease as policies become more permissive, while intent drift remains fixed because it is computed against a policy-invariant baseline.

The analysis also shows that violations are concentrated around a small number of destination ports, with port 25565 emerging as the dominant outlier. This supports service-aware policy refinement and micro-segmentation as practical mechanisms for improving intent assurance. Future work will validate the framework on additional datasets, explore adaptive drift baselines, and extend policy predicates to include richer context such as source identity, destination role, protocol behavior, and temporal patterns.

Acknowledgment

The research was partially funded by NSF under the CICI: TCR program, IRIS: Instrumentation for Research and Inter-institutional SOC, Award Number NSF 2319793.

References

[1] T. Haikal, E. Hammad, and S. Ismail, "Characterizing large-scale adversarial activities through large-scale honey-nets," arXiv preprint arXiv:2512.06557, 2025, Accepted at IEEE UEMCON 2025. DOI: 10.48550/arXiv.2512.06557.

[2] E. Zeydan and Y. Turk, "Recent advances in intent-based networking: A survey," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, IEEE, 2020, pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9128913.

[3] A. Leivadeas and M. Falkner, "A survey on intent-based networking," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 625–655, 2022. DOI: 10.1109/COMST.2022.3215919.

[4] T. A. Khan, A. Muhammad, K. Abbas, and W.-C. Song, "Intent-based networking platform: An automated approach for policy and configuration of next-generation networks," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing (SAC '21)*, 2021, pp. 1921–1930. DOI: 10.1145/3412841.3442064.

[5] A. Mekrache, A. Ksentini, and C. Verikoukis, "Intent-based management of next-generation networks: An llm-centric approach," IEEE Network, vol. PP, no. 99, pp. 1–1, 2024. DOI: 10.1109/MNET.2024.3420120.

[6] J. Kim, H. Okhravi, D. J. Tian, and B. E. Ujcich, "Security challenges of intent-based networking," Communications of the ACM, vol. 67, no. 7, pp. 56–65, 2024. DOI: 10.1145/3639702.

[7] J. Andrade-Hoz, Q. Wang, and J. M. Alcaraz-Calero, "Infrastructure-wide and intent-based networking dataset for 5g-and-beyond ai-driven autonomous networks," Sensors, vol. 24, no. 3, p. 783, 2024. DOI: 10.3390/s24030783.

[8] S. Ismail, S. Dandan, and M. King, "A lightweight machine learning approach for anomalous unsolicited network traffic detection by observing network telescopes," in *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, 2025.

[9] S. Ismail, S. Dandan, and M. King, "Understanding honeypots: Observing malicious activities over telnet," in *2025 IEEE International Conference on Electro Information Technology (eIT)*, 2025, pp. 167–172. DOI: 10.1109/eIT64391.2025.11103659.