INTACT: Intent-Aware Representation Learning for Cryptographic Traffic Violation Detection

Rahul D Ray
BITS Pilani, Hyderabad Campus
Hyderabad, India
f20242213@hyderabad.bits-pilani.ac.in

ABSTRACT

Security monitoring in Critical National Infrastructure systems such as energy grids, telecommunications backbones, and public service networks requires reliable enforcement of explicit operational policies under evolving threat conditions. Conventional anomaly detection methods treat violations as statistical deviations from observed data distributions. In cryptographic traffic monitoring, however, violations are defined not by rarity but by formal policy constraints including key reuse prohibition, downgrade prevention, and bounded key lifetimes. This mismatch limits interpretability, policy alignment, and adaptability in safety critical environments.We introduce INTACT, an intent aware representation learning framework that reformulates violation detection as conditional constraint learning. Instead of learning a static anomaly boundary over behavioral features, INTACT models the probability of violation conditioned jointly on observed behavior and declared security intent. The architecture factorizes representation learning into behavioral and intent encoders whose fused embeddings yield a policy parameterized family of decision boundaries, enabling explicit alignment between detection logic and operational constraints. We evaluate INTACT on a large scale real world network flow dataset and a 210000 trace synthetic multi intent cryptographic corpus modeling controlled policy violations and distribution shift. INTACT matches or exceeds strong unsupervised and supervised baselines, achieving near perfect discrimination with AUROC up to 1.0000 in real traffic and consistently superior performance on relational and composite violations in synthetic settings. These results demonstrate that explicit intent conditioning improves discrimination, interpretability, and robustness, properties essential for trustworthy AI deployment in critical infrastructure environments.

KEYWORDS

Intent-Aware Learning, Cryptographic Traffic Analysis, Policy-Conditioned Detection, Critical National Infrastructure, AI Safety, Multi-Intent Anomaly Detection, Distribution Shift Robustness

1 INTRODUCTION

The rapid growth of encrypted network communication has transformed intrusion detection, especially in Critical National Infrastructure systems such as energy grids, telecommunications backbones, transportation networks, and public service platforms. Because SSL/TLS and VPN dominated environments render deep packet inspection ineffective, machine learning methods based on flow level statistics have become central. Deep architectures achieve strong performance in encrypted traffic classification and anomaly detection, including CNN–GRU hybrids [1], CNN–LSTM–autoencoder frameworks [20], multilayer autoencoders [19], self supervised contrastive learning [12], and error resilient recurrent networks [24]. GAN based systems [22], reinforcement learning approaches [18], explainable IDS frameworks [9, 11], and privacy preserving intrusion mechanisms [2, 7] further extend this paradigm.

Despite these advances, the dominant formulation remains statistical anomaly detection. Models distinguish normal from abnormal behavior via deviations in learned representations, and surveys confirm that most encrypted traffic detection relies on supervised or unsupervised representation learning to approximate data distributions [6, 10, 14]. Violations are therefore treated as statistical outliers.

In practice, safety critical infrastructure operates under explicit policy frameworks such as intent based networking, compliance monitoring, and formal policy validation. Intent based assurance systems validate declared intents using monitoring and learning [3–5, 16]. Formal specification based detection [13], first order policy validation [15], policy aware intrusion systems [17], compliance architectures [8], and privacy preserving policies [7] emphasize constraint satisfaction rather than density estimation. Earlier work on user intention based detection [21], safety violation analysis [23], and encrypted policy aware intrusion [2] similarly highlights constraint driven monitoring.

This exposes a conceptual gap. Existing encrypted traffic models estimate

$$g(x) \approx P(y = 1 \mid x),$$

treating anomaly as intrinsic to behavior x. In policy driven systems, however, violations depend on declared constraints. A flow may be compliant under one lifetime threshold yet violating under another, and downgrade or key reuse events may be acceptable or prohibited depending on policy.

We introduce **INTACT (INTent-Aware Cryptographic Traffic)**, which reformulates violation detection as conditional constraint learning,

$$f(x, z) \approx P(y = 1 \mid x, z),$$

where z encodes declared security intent such as lifetime thresholds, downgrade prohibitions, and reuse constraints. By factorizing learning into behavioral and intent encoders, INTACT produces policy parameterized decision manifolds rather than a fixed anomaly boundary. Unlike prior frameworks [1, 12, 19, 20, 22, 24], it embeds constraint semantics directly into representation space, aligning with specification based detection [13], policy validation [15], and intent aware assurance [4, 16].

2nd Workshop on AI for Critical National Infrastructure (AI4CNI-26) @ AAMAS 2026, F.L. da Silva, W. Su, R. Glatt, C. Schroeder de Witt, May 25 – 29, 2026, Paphos, Cyprus.

We evaluate INTACT on a real world network flow dataset and a synthetic multi intent cryptographic corpus. Across violation types and distribution shift scenarios, it improves discrimination, interpretability, and robustness relative to strong supervised and unsupervised baselines. This work identifies the mismatch between statistical anomaly detection and constraint based monitoring, proposes a policy conditioned representation learning framework, introduces a dual dataset evaluation, and demonstrates that intent conditioning strengthens robustness in safety critical environments.

2 DATASET PREPARATION AND EXPERIMENTAL PROTOCOL

This study employs a dual dataset strategy combining large scale real world network traffic with controlled synthetic cryptographic traces. The real world dataset provides ecological validity through natural temporal dynamics and diverse attack scenarios observed in operational environments, while the synthetic dataset enables precise control over violation mechanisms and full ground truth visibility. Together, they form a comprehensive testbed for evaluating intent aware detection in safety critical and policy driven infrastructure contexts.

2.1 Real-World Network Flow Dataset

The real-world dataset comprises multiple days of enterprise network traffic containing benign activity and diverse attack campaigns, including distributed denial-of-service, port scanning, web-based exploits, and infiltration attempts. All daily captures were aggregated into a unified corpus of 2,830,743 structured flow records, each described by dozens of statistical features and a ground-truth traffic label. The primary objective is reformulated as binary intent-based detection: identifying abnormal flow lifetimes defined by excessive duration relative to normal operational behavior.

2.1.1 Data Cleaning and Feature Selection. A compact subset of behaviorally meaningful features was selected to capture core traffic characteristics:

- Total flow duration
- Number of packets transmitted in forward and backward directions
- Mean packet size in each direction
- Throughput intensity (bytes per second and packets per second)

These features describe temporal persistence, volumetric intensity, and directional asymmetry—attributes particularly relevant for detecting abnormal long-lived connections. All high-dimensional or protocol-specific attributes were removed to reduce dimensionality and mitigate spurious correlations, yielding a compact, interpretable feature space.

Numerical instabilities arising from rate computations were addressed by replacing infinite values with missing entries and subsequently removing all records containing missing values. Post-cleaning inspection confirmed the absence of missing or infinite values, ensuring numerical stability during normalization and training.

2.1.2 Temporal Ordering and Data Partitioning. Network traffic is inherently temporal; random shuffling would leak future information and inflate performance. Therefore, the dataset was ordered chronologically and partitioned strictly by time:

- 60% earliest records for training (1,696,725 samples)
- 20% for validation (565,575 samples)
- 20% for testing (565,576 samples)

This simulates a realistic deployment where models are trained on historical data and evaluated on future, unseen traffic.

2.1.3 Definition of Lifetime Violation. Rather than using attack taxonomy labels, a statistical threshold was derived from normal behavior. Using only benign flows from the training partition, the empirical distribution of flow duration was analyzed, and the 95th percentile was selected as the cutoff. Any flow exceeding this threshold is labeled a violation (1), while others are normal (0). The threshold value (113,046,291 in dataset time units) was computed exclusively from the training set to prevent leakage.

Class distributions across partitions are:

- Training: 1,621,064 normal, 75,661 violations
- Validation: 536,648 normal, 28,927 violations
- Test: 551,819 normal, 13,757 violations

Moderate class imbalance exists, but both classes are well represented, ensuring learnability and robust assessment.

2.1.4 Feature Normalization. Features span different scales (e.g., duration in millions, packet counts in tens). To prevent high-magnitude features from dominating optimization, standard score normalization was applied:

$$z = \frac{x - \mu}{\sigma}, \tag{1}$$

with μ and σ computed exclusively from the training partition and then applied unchanged to validation, test, and shifted sets. Verification confirmed training means near zero and standard deviations near one, with minor expected deviations in other splits—confirming proper non-leaking normalization.

2.1.5 Construction of Controlled Distribution Shift. To evaluate robustness under covariate shift, two modified test variants were created by scaling flow duration values (doubled and tripled) while keeping all other features and violation labels unchanged. This isolates feature distribution shift from label shift. The Kolmogorov–Smirnov test confirmed significant divergence from the original test distribution (KS statistics 0.2167 and 0.2566, $p \approx 0$), establishing meaningful out-of-distribution evaluation settings.

2.2 Synthetic Intent-Anomaly Dataset

Real intrusion datasets lack fine-grained control over specific security intent violations such as key reuse, algorithm downgrade, or lifetime exceedance. To complement the real data, we constructed a large-scale synthetic corpus of cryptographic operation traces with explicitly controlled violation mechanisms and precise ground truth. The dataset contains 210,000 distinct traces comprising 4,193,768 individual operation records, each representing a sequence of cryptographic events with explicit key lifecycle semantics and algorithmic properties.

Table 1: Comparative summary of real and synthetic datasets.

Section	Real Network Flow Dataset	Synthetic Intent-Anomaly Dataset
A. Overall Scale		
Total records / traces	2,830,743 flow records	210,000 traces
Total atomic events	2,830,743	4,193,768 operations
Input dimensionality	7 numerical features	8 per-operation attributes
Target variables	Binary lifetime violation	Reuse, Downgrade, Lifetime flags
Data source	Real network traffic (multi-day)	Controlled generative process
Temporal structure	Chronological ordering	Within-trace temporal generation
Scaling applied	Standardization (train statistics)	Stochastic noise + parametric scaling
B. Data Partitioning		
Training set size	1,696,725 (60%)	Unified corpus (no fixed split)
Validation set size	565,575 (20%)	Experiment-dependent subsets
Test set size	565,576 (20%)	Per-trace evaluation
Temporal split	Yes (strict chronological)	Not required
C. Violation Distribution		
Training violations	75,661	Controlled by construction
Training normal samples	1,621,064	120,000 normal traces
Validation violations	28,927	—
Test violations	13,757	—
Reuse-only traces	Not applicable	20,000
Downgrade-only traces	Not applicable	20,000
Lifetime-only traces	Derived statistically	30,000
Composite violations	Not labeled separately	20,000
D. Distribution Shift Evaluation		
Shift Variant 1	Duration ×2	Scale factor 1.5 (10,000 traces)
Shift Variant 2	Duration ×3	Scale factor 3.0 (10,000 traces)
Statistical verification	KS test confirmed shift	Structural parameter scaling
E. Class Imbalance		
Violation rate (train)	∼4.5%	Controlled design (∼balanced)
Label derivation	95th percentile threshold	Deterministic rule computation

2.2.1 Generative Assumptions and Operational Model. Each trace models a sequence of operations drawn from a finite vocabulary: key generation, encryption, decryption, signing, and verification. The number of operations per trace is stochastic (Poisson-like, truncated), producing realistic length variability. Inter-operation timing follows an exponential distribution (memoryless arrival), and operation durations are drawn from a log-normal distribution (positively skewed, multiplicative).

Each generated key is assigned:

- A creation timestamp
- A finite lifetime from a bounded continuous range
- A cryptographic strength parameter (strong ≥256-bit security; weak below threshold)

Internal logical consistency is enforced: operations can only reference valid keys; if none exists, a new key is generated. Baseline traces thus represent valid cryptographic behavior.

2.2.2 Noise Injection and Distribution Matching. Controlled multiplicative Gaussian noise is applied to inter-arrival intervals and operation durations to introduce natural variability while preserving structural semantics. Timestamps are recomputed cumulatively to maintain monotonicity, preventing models from exploiting deterministic generative artifacts.

2.2.3 Controlled Violation Mechanisms. Three independent violation dimensions are encoded:

Key Lifetime Violation: A target key's final operation is shifted past its expiration, with subsequent timestamps adjusted to preserve order.

Algorithm Downgrade Violation: Strong algorithms are replaced with weak ones across a trace.

Key Reuse Violation: Two distinct traces share a common key identifier (key remains valid during reuse), modeling improper cross-context sharing.

Violations can be applied independently or in combination to create composite scenarios.

2.2.4 Dataset Composition. Traces are generated across eight categories:

- Normal (no violations)
- Reuse-only, Downgrade-only, Lifetime-only
- Reuse+Downgrade, Reuse+Lifetime, Downgrade+Lifetime
- Reuse+Downgrade+Lifetime

Generation counts approximate balanced representation while maintaining a dominant normal class:

- 120,000 normal traces
- 20,000 reuse-only
- 20,000 downgrade-only
- 30,000 lifetime-only
- 20,000 composite traces (total 210,000)

2.2.5 Expansion to Tabular Representation. Each trace is expanded to a flat table where each row corresponds to one operation, containing:

- Trace identifier, step index, timestamp
- Operation type, key identifier, algorithm identifier
- Assigned key lifetime, operation duration

The final expanded dataset has 4,193,768 rows and 8 columns.

2.2.6 Automatic Violation Annotation. Violation labels are computed deterministically from the generated data:

- Lifetime violations: timestamp, key creation + lifetime
- Downgrade violations: algorithm strength below threshold
- Reuse violations: key appears in multiple traces

This ensures internal consistency between generative logic and annotation. Verification confirmed that category counts align closely with generation targets.

2.2.7 Distribution Shift Construction. Two shifted synthetic datasets were generated by scaling temporal characteristics and key lifetimes by factors 1.5× and 3.0×. Each contains 10,000 traces (200,000 operations). The structural generation process is unchanged, but temporal dynamics differ significantly, enabling robustness evaluation under changes in arrival rates, key lifetime ranges, and event density—mirroring realistic operational variations.

2.2.8 Role in Experimental Framework. The synthetic dataset serves three purposes: controlled evaluation of intent-aware detection, analysis of violation separability under known generative conditions, and robustness testing under synthetic distribution shifts. It provides full ground-truth visibility into key lifecycle semantics and algorithmic properties, enabling precise attribution of model behavior to specific violation mechanisms—complementing the ecological validity of the real-world data.

3 FORMALIZATION OF INTENT-CONDITIONED VIOLATION DETECTION

Let $x \in \mathbb{R}^d$ denote a behavioral representation of cryptographic activity and $z \in \mathbb{R}^k$ encode declared security intent (e.g., lifetime thresholds or structured policy constraints). Let $y \in \{0, 1\}$ indicate violation of intent z.

Classical anomaly detection models estimate

$$g(x) \approx \mathbb{P}(y = 1 \mid x),$$

implicitly assuming violations are intrinsic to behavior. In policy-driven systems, however, violations are defined relative to declared constraints. We therefore model

$$f(x, z) = \mathbb{P}(y = 1 \mid x, z),$$

which induces a policy-parameterized family of decision manifolds

$$\mathcal{M}(z) = \{x \in \mathbb{R}^d : f(x, z) = 0.5\}.$$

For scalar thresholds $z = \tau$, suppose violations arise from $y = \mathbf{1}(g(x) > \tau)$. INTACT learns a differentiable relaxation:

$$f(x, \tau) = \sigma\big(g_\theta(x) - h_\theta(\tau)\big),$$

implementing a learnable comparator between behavioral score and policy magnitude.

We parameterize f via factorization:

$$f(x, z) = \sigma\big(\phi\big(\psi_b(x), \psi_z(z)\big)\big),$$

where $\psi_b : \mathbb{R}^d \to \mathbb{R}^m$ encodes behavior, $\psi_z : \mathbb{R}^k \to \mathbb{R}^n$ encodes intent, and ϕ fuses both representations. With ReLU activations,

$$\psi_b(x) = W_3\rho(W_2\rho(W_1x + b_1) + b_2) + b_3,$$

$$\psi_z(z) = V_2\rho(V_1z + c_1) + c_2.$$

Concatenation $h = [\psi_b(x); \psi_z(z)]$ is mapped through a nonlinear layer

$$\phi(h) = u^\top\rho(W_4h + b_4) + b_5,$$

yielding $f(x, z) = \sigma(\phi(h))$. Nonlinear coupling enables intent-conditioned feature reweighting, threshold comparison, and relational interactions.

Training minimizes binary cross-entropy:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\big[y_i \log f_\theta(x_i, z_i) + (1 - y_i)\log(1 - f_\theta(x_i, z_i))\big].$$

Gradients propagate through both encoders,

$$\nabla_\theta\phi = \frac{\partial\phi}{\partial h_b}\nabla_\theta\psi_b + \frac{\partial\phi}{\partial h_z}\nabla_\theta\psi_z,$$

ensuring that representation learning is jointly shaped by behavioral signals and constraint semantics.

Conceptually, this enlarges the hypothesis space from $g : \mathbb{R}^d \to [0, 1]$ to $f : \mathbb{R}^{d+k} \to [0, 1]$, while controlling complexity through low-dimensional embeddings $m \ll d$, $n \ll k$. Decision boundaries become smooth families indexed by policy, with sensitivity

$$\frac{\partial f}{\partial z} = \sigma'(\phi)\frac{\partial\phi}{\partial h_z}\frac{\partial\psi_z}{\partial z}.$$

Unlike density-based methods that approximate $p(x)$, INTACT directly models $\mathbb{P}(y = 1 \mid x, z)$, reframing anomaly detection as policy-conditioned constraint evaluation rather than statistical rarity estimation.

4 INTACT: INTENT-AWARE CRYPTOGRAPHIC TRAFFIC

Cryptographic network traffic is governed by explicit security policies: keys must not be reused, deprecated algorithms must not be negotiated, and key lifetimes must not exceed predefined thresholds. These policies define formal intent constraints over observable traffic behavior. Conventional anomaly detection systems do not explicitly model such intent; instead, they estimate statistical normality from data and flag deviations. This statistical framing implicitly assumes that violations correspond to low-density regions of feature space. However, in security contexts, violations are not necessarily rare—they are defined relative to declared constraints.

We therefore reformulate cryptographic anomaly detection as an intent-conditioned inference problem. Let $x \in \mathbb{R}^d$ denote a behavioral representation extracted from traffic (flow-level or trace-level features). Let z denote a structured encoding of security intent (e.g., lifetime threshold, reuse prohibition, downgrade constraint). The objective is not to model $p(x)$, but to estimate

$$f(x, z) = \mathbb{P}(\text{violation} \mid x, z). \tag{2}$$

Under this formulation, anomaly detection becomes conditional violation detection. The same behavioral pattern may or may not constitute a violation depending on the declared policy parameter z. This explicit conditioning is the conceptual core of INTACT.

4.1 Architectural Design

INTACT is implemented as a dual-branch neural architecture that separately encodes behavioral signals and intent semantics before fusing them for violation prediction.

4.1.1 Behavioral Encoder. The behavioral branch processes observable traffic features derived from cryptographic flows or aggregated traces. For the real-world dataset, the input dimensionality is seven standardized flow-level statistics. For the synthetic dataset, the input consists of seventeen aggregated trace-level attributes capturing algorithm selection, key identifiers, duration statistics, and operational counts.

The behavioral encoder consists of a sequence of fully connected layers with nonlinear activation functions. In the real-data configuration:

- Dense layer with 128 units (ReLU)
- Dense layer with 64 units (ReLU)
- Dense layer with 32 units (ReLU)

This produces a 32-dimensional behavioral embedding h_b, which captures compact representations of traffic semantics while preserving nonlinear feature interactions.

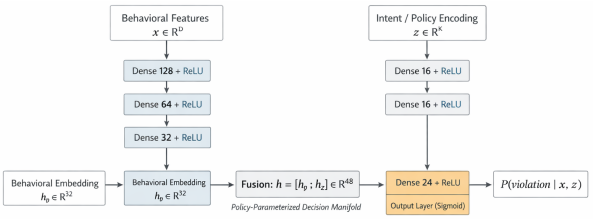


Figure 1: INTACT architecture: behavioral encoder, intent encoder, fusion layer, and violation prediction output.

4.1.2 Intent Encoder. The intent branch encodes structured policy information. In the real dataset, intent corresponds to the scaled lifetime threshold computed as the 95th percentile of benign flow duration. In the synthetic dataset, intent corresponds to one of the three violation categories (reuse, downgrade, lifetime), represented through structured inputs.

The intent encoder transforms this low-dimensional policy input into a learnable semantic embedding through:

- Dense layer with 16 units (ReLU)
- Dense layer with 16 units (ReLU)

This yields a 16-dimensional intent embedding h_z, enabling the model to internalize the semantics of the constraint rather than treating it as a raw scalar.

4.1.3 Fusion and Decision Layer. The behavioral embedding $h_b \in \mathbb{R}^{32}$ and intent embedding $h_z \in \mathbb{R}^{16}$ are concatenated into a 48-dimensional joint representation:

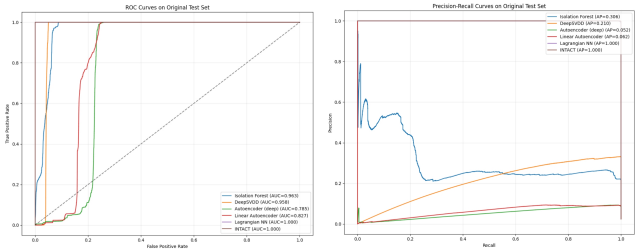
$$h = [h_b; h_z]. \tag{3}$$

This fused representation is passed through:

- Dense layer with 24 units (ReLU)
- Output layer with 1 unit (Sigmoid)

The final output represents the probability that behavior x violates intent z.

The real-data configuration contains 12,865 trainable parameters. The architecture remains lightweight while providing sufficient capacity to model nonlinear interactions between traffic behavior and policy constraints.



(a) ROC curve on the real dataset test set. **(b) Precision–Recall curve on the real dataset test set.**

Figure 2: Performance evaluation on the real-world network flow dataset. Left: ROC curve on the test set. Right: Precision–Recall curve on the test set.

4.2 Learning Objective

The model is trained using binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log f(x_i, z_i) + (1 - y_i)\log(1 - f(x_i, z_i))\right]. \tag{4}$$

For unsupervised baselines, only normal samples are used for training. In contrast, INTACT leverages labeled supervision, allowing it to directly learn violation manifolds conditioned on intent.

Early stopping is applied based on validation AUC, and threshold selection is performed by maximizing validation F1-score derived from the precision–recall curve.

4.3 Distinction from Conventional Architectures

The novelty of INTACT does not lie solely in its layer composition. Feedforward architectures are well established. The key innovation is the structural separation of behavior encoding and policy encoding within a unified model. This design transforms violation detection from a density estimation problem into a conditional constraint evaluation problem.

Conventional supervised classifiers learn a static mapping $x \mapsto y$. INTACT learns a conditional mapping $(x, z) \mapsto y$. This subtle reformulation allows the same network to adapt to multiple policy definitions without retraining independent models or redefining anomaly thresholds externally.

5 COMPREHENSIVE MODEL EVALUATION ACROSS REAL-WORLD AND SYNTHETIC DATASETS

To rigorously assess the effectiveness of the proposed intent-conditioned architecture and establish strong comparative baselines, a comprehensive benchmarking study was conducted across both the real-world network flow dataset and the synthetic multi-intent cryptographic dataset described previously. The evaluation protocol was unified across datasets to ensure methodological consistency while respecting structural differences between single-intent and multi-intent settings.

All models were trained using temporally or structurally partitioned training sets, validated on held-out validation sets for

Table 2: Architectural Specifications of All Evaluated Models

Model	Learning Paradigm	Input Dim.	Architecture	Embedding Size	Output	Params
Nonlinear AE	Unsupervised	7 / 17	Enc: 16→8 (ReLU); Dec: 16→linear	8	Recon. error	∼1–3K
Linear AE	Unsupervised	7 / 17	Enc: 4 (linear); Dec: linear	4	Recon. error	<1K
Deep SVDD	Unsupervised	7 / 17	Embedding network + hypersphere constraint	Latent vector	Distance score	Few K
Isolation Forest	Unsupervised	7 / 17	100-tree ensemble (random partitions)	N/A	Path-length score	Tree ensemble
Supervised NN	Supervised	7 / 17	64→32→16 (ReLU)→1 (Sigmoid)	64–32–16	Binary prob.	∼3K–5K
INTACT	Supervised (Multi-input)	Behavior: 7 / 17; Intent: 1	Behavior: 128→64→32; Intent: 16→16; Fusion: 24→1	32 + 16 → 48	Binary prob.	12,865

threshold optimization, and finally evaluated on in-distribution test partitions as well as distribution-shift variants. Performance was quantified using Area Under the Receiver Operating Characteristic Curve (AUROC), Area Under the Precision–Recall Curve (AUPRC), and F1-score at a validation-optimized threshold derived from the precision–recall curve. Unsupervised models were trained exclusively on normal samples. Supervised models were trained using full label supervision. All neural architectures were trained with GPU acceleration (Tesla P100, 16GB), ensuring stable optimization over large-scale data.

5.1 Evaluation on the Real-World Network Flow Dataset

The real-world dataset contains temporally ordered network flows with a single violation label corresponding to lifetime threshold exceedance. Unsupervised methods were trained on 1,621,064 normal flows from the training partition. Two distribution-shift test variants were constructed by scaling flow duration, enabling covariate shift robustness analysis.

5.1.1 Unsupervised Methods. A nonlinear autoencoder was trained for 20 epochs using reconstruction error as the anomaly score. Validation performance reached AUROC = 0.8964 and AUPRC = 0.1890. On the in-distribution test set, AUROC decreased to 0.7854 with AUPRC = 0.0518 and F1 = 0.1687. Under duration scaling shifts, AUROC increased substantially (0.9024 and 0.9859), while F1 remained low (∼0.16), indicating inflated separability but poor calibration under magnitude perturbations.

Deep SVDD, trained for 20 epochs on normal flows, achieved validation AUROC = 0.9776 and AUPRC = 0.4928. On the test set, AUROC = 0.9575 and AUPRC = 0.2101 with F1 = 0.4963. Under shift, AUROC remained stable (0.9650 and 0.9762) while F1 declined moderately (∼0.32), demonstrating strong compactness of normal representations.

Isolation Forest (100 trees, 5% contamination) achieved validation AUROC = 0.9732 and AUPRC = 0.4972. Test performance reached AUROC = 0.9631 and AUPRC = 0.3058 with F1 = 0.4115. Under shift, AUROC remained above 0.95 with modest F1 degradation, indicating robust tree-based partitioning under covariate scaling.

5.1.2 Supervised Neural Baselines. A fully supervised feedforward network achieved perfect validation performance (AUROC = 1.0000, AUPRC = 1.0000), confirming near-deterministic separability. The optimal threshold was 0.3670. This suggests that lifetime violations in this dataset are strongly governed by duration-related features.

A linear autoencoder with 4-dimensional bottleneck achieved validation AUROC = 0.9919 and AUPRC = 0.8556. On test data, AUROC = 0.9352 and AUPRC = 0.1598 with F1 = 0.2594. Under extreme scaling, AUROC approached 0.999, again reflecting exaggerated reconstruction deviation under magnitude inflation.

5.1.3 Intent-Conditioned Architecture. The intent-conditioned architecture was evaluated using the lifetime threshold computed as the 95th percentile of benign flow duration (scaled value 2.2134). The model contains 12,865 trainable parameters and processes behavioral features and threshold intent through separate encoders before fusion.

Validation performance reached AUROC = 1.0000 and AUPRC = 1.0000 with optimal threshold 0.8614. On the in-distribution test set, performance was near-perfect: AUROC = 1.0000, AUPRC = 1.0000, F1 = 0.9973. Under shift, AUROC remained high (0.9702 and 0.9626), while AUPRC declined (0.2952 and 0.2497) due to calibration sensitivity.

Overall, lifetime violation detection in real flow data appears structurally simple and heavily duration-driven. Supervised and intent-conditioned models achieve near-deterministic performance.

5.2 Evaluation on the Synthetic Multi-Intent Dataset

The synthetic dataset introduces a substantially more complex setting with 210,000 traces (4,193,768 operations aggregated to 17 features per trace) and three independent violation intents: reuse, downgrade, and lifetime. The label matrix has dimensionality (210,000 × 3). The dataset was split into 167,964 training, 21,047 validation, and 20,989 test traces. Positive class proportions remain stable across splits: reuse and downgrade ∼16.6%, lifetime ∼13.9%.

Two shift variants were constructed but contain only normal traces, leading to undefined ROC/AUPRC metrics under those subsets.

5.2.1 Unsupervised Methods. Isolation Forest exhibited weak performance for reuse (AUROC = 0.5057) and moderate performance for downgrade (0.6757) and lifetime (0.7074). Deep SVDD showed strong downgrade discrimination (AUROC = 0.8332) but weak reuse (0.5367) and moderate lifetime (0.6091). The nonlinear autoencoder achieved AUROC = 0.7261 for downgrade, 0.6070 for reuse, and near-random 0.5014 for lifetime. The linear autoencoder performed poorly on reuse and downgrade but achieved strong lifetime detection (AUROC = 0.8483).

These results demonstrate heterogeneous separability across intents. Downgrade violations introduce direct feature-level deviations and are relatively easy for geometric anomaly detectors. Lifetime violations exhibit partial linear separability. Reuse violations are inherently relational across traces and are not well captured by magnitude-based anomaly scoring.

Table 3: Unified Performance Comparison Across Real and Synthetic Datasets

Dataset	Intent	Model	AUROC	AUPRC	F1
Real (Test)	Lifetime	Nonlinear Autoencoder	0.7854	0.0518	0.1687
		Deep SVDD	0.9575	0.2101	0.4963
		Isolation Forest	0.9631	0.3058	0.4115
		Linear Autoencoder	0.9352	0.1598	0.2594
		Supervised NN	∼1.0000	∼1.0000	∼1.0000
		INTACT	1.0000	1.0000	0.9973
Real (Shift2)	Lifetime	Nonlinear Autoencoder	0.9024	0.1054	0.1661
		Deep SVDD	0.9650	0.2432	0.3224
		Isolation Forest	0.9547	0.2546	0.3419
		Linear Autoencoder	0.9986	0.9406	0.2385
		INTACT	0.9702	0.2952	0.4047
Real (Shift3)	Lifetime	Nonlinear Autoencoder	0.9859	0.6390	0.1638
		Deep SVDD	0.9762	0.3099	0.3123
		Isolation Forest	0.9534	0.2477	0.3374
		Linear Autoencoder	0.9991	0.9632	0.2100
		INTACT	0.9626	0.2497	0.3901
Synthetic (Test)	Reuse	Isolation Forest	0.5057	0.1618	0.2843
		Deep SVDD	0.5367	0.1764	0.2845
		Nonlinear Autoencoder	0.6070	0.2475	0.3165
		Linear Autoencoder	0.4849	0.1547	0.2843
		Supervised NN	0.7793	0.6471	0.5939
		INTACT	0.7820	0.6515	0.5963
	Downgrade	Isolation Forest	0.6757	0.2304	0.3656
		Deep SVDD	0.8332	0.3829	0.5131
		Nonlinear Autoencoder	0.7261	0.2824	0.4109
		Linear Autoencoder	0.5191	0.1670	0.2923
		Supervised NN	1.0000	1.0000	0.9999
		INTACT	1.0000	1.0000	0.9999
	Lifetime	Isolation Forest	0.7074	0.2383	0.3390
		Deep SVDD	0.6091	0.1786	0.2792
		Nonlinear Autoencoder	0.5014	0.1384	0.2494
		Linear Autoencoder	0.8483	0.4958	0.5641
		Supervised NN	0.9797	0.8199	0.8493
		INTACT	0.9797	0.8210	0.8486

5.2.2 Supervised Neural Baseline. The fully supervised neural classifier significantly improved performance. For reuse, AUROC = 0.7793 and AUPRC = 0.6471. For lifetime, AUROC = 0.9797 and AUPRC = 0.8199. For downgrade, performance was perfect (AUROC = 1.0000). This confirms that downgrade and lifetime violations are highly separable under supervision, while reuse remains the most challenging.

5.2.3 Intent-Conditioned Architecture (INTACT). The intent-conditioned model achieved AUROC = 0.7820 (reuse), 0.9797 (lifetime), and 1.0000 (downgrade). AUPRC values were 0.6515, 0.8210, and 1.0000 respectively. Performance closely matches or slightly exceeds the supervised baseline.

The key distinction lies in representation structure: conditioning explicitly aligns decision boundaries with declared intent rather than implicitly learning separate classifiers. Importantly, reuse detection—difficult for unsupervised methods—remains strong under intent conditioning.

6 DISCUSSION

Across both datasets, violation separability is driven more by generative structure than model capacity. Real-world lifetime violations are nearly threshold-deterministic and thus highly separable under supervision, whereas synthetic reuse violations require modeling cross-trace relational dependencies. Unsupervised methods capture magnitude-based deviations but struggle with relational semantics. Reconstruction-based models inflate AUROC under covariate scaling while degrading calibrated F1-scores, showing that rank-based separability does not ensure operational reliability. Under distribution shift, AUROC often remains high while AUPRC and F1 decline, revealing threshold instability.

Supervised baselines outperform unsupervised approaches, particularly in the real dataset where lifetime violations are duration-driven. However, these models learn fixed decision boundaries tied to training statistics and do not explicitly encode policy parameters. INTACT matches or exceeds these baselines while preserving interpretability through intent conditioning. In the synthetic setting, where multi-intent interactions produce heterogeneous violation geometries, conditional modeling is especially beneficial. Reuse violations cannot be reduced to simple magnitude thresholds yet remain detectable through intent-aware representations. Conceptually, the framework models $f(x, z) = \mathbb{P}(y = 1 \mid x, z)$, treating policy as a first-class input and parameterizing decision boundaries by intent to enable semantic alignment and adaptivity under policy drift.

In Critical National Infrastructure, monitoring is distributed across sensors, gateways, and control nodes. INTACT supports multi-agent deployment in which local agents compute behavioral embeddings $\psi_b(x)$ while receiving shared policy embeddings $\psi_z(z)$

from a central authority. Each agent outputs a local violation probability that can be aggregated through weighted averaging, attention mechanisms, or hierarchical fusion to capture cross-site correlations. Agents may exchange compressed embeddings rather than raw traffic, enabling privacy-preserving and bandwidth-efficient coordination aligned with distributed safety assurance.

A limitation of the real-world evaluation is that lifetime violations are defined by a 95th percentile threshold computed from benign training data. Although this avoids leakage and preserves temporal realism, it simplifies the task because violations are largely duration-driven. Near-perfect discrimination on this dataset should therefore not be interpreted as universal complexity. The synthetic multi-intent corpus provides a stricter test through controlled reuse, downgrade, and composite violations that are not purely threshold-based. Calibration under joint behavior–policy shift and scalable relational reasoning across distributed agents remain open challenges, but conditional constraint learning offers a principled foundation for policy-driven cryptographic monitoring in safety-critical infrastructure systems.

7 CONCLUSION

We presented INTACT, an intent aware representation learning framework for cryptographic traffic monitoring in Critical National Infrastructure settings. The method reformulates violation detection as conditional constraint learning, replacing static anomaly boundaries with policy parameterized decision surfaces aligned with declared security intent. By factorizing behavioral and intent encoders within a unified network, INTACT enables differentiable constraint comparison and multi intent modeling while maintaining interpretability and computational efficiency. Evaluation on a large scale real world network flow dataset with 2.8 million flows and a synthetic multi intent corpus with 210000 traces and 4.2 million operations shows that INTACT matches or exceeds strong supervised and unsupervised baselines, achieving near perfect discrimination for lifetime violations and strong performance on relational and composite violations such as key reuse. Under controlled distribution shift, discrimination remains robust, though calibration stability remains important for safety critical deployment. Conceptually, the framework models violation inference as learning a differentiable constraint operator $f(x, z)$ rather than estimating densities or static classifiers, enabling semantic alignment with operational policies and structural adaptivity to evolving infrastructure requirements. These properties position INTACT as a principled foundation for policy aligned and trustworthy AI based monitoring in critical infrastructure environments.

REFERENCES

[1] Taimur Bakhshi and Bogdan Ghita. 2021. Anomaly detection in encrypted internet traffic using hybrid deep learning. *Security and Communication Networks* 2021, 1 (2021), 5363750.

[2] Sébastien Canard, Aïda Diop, Nizar Kheir, Marie Paindavoine, and Mohamed Sabt. 2017. BlindIDS: Market-compliant and privacy-friendly intrusion detection system over encrypted traffic. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 561–574.

[3] Federica De Trizio, Giancarlo Sciddurlo, Dominga Rutigliano, Giuseppe Piro, and Gennaro Boggia. 2024. A novel malicious intent detection approach in intent-based enterprise networks. In *2024 20th International Conference on Network and Service Management (CNSM)*. IEEE, 1–7.

[4] Molka Gharbaoui, Filippo Sciarrone, Mattia Fontana, Piero Castoldi, and Barbara Martini. 2026. Assurance and Conflict Detection in Intent-Based Networking: A Comprehensive Survey and Insights on Standards and Open-Source Tools. *IEEE Transactions on Network and Service Management* 23 (2026), 1891–1912.

[5] Urslla Uchechi Izuazu, Mounir Bensalem, and Admela Jukan. 2025. A Secured Intent-Based Networking (sIBN) with Data-Driven Time-Aware Intrusion Detection. *arXiv preprint arXiv:2511.05133* (2025).

[6] Il Hwan Ji, Ju Hyeon Lee, Min Ji Kang, Woo Jin Park, Seung Ho Jeon, and Jung Taek Seo. 2024. Artificial intelligence-based anomaly detection technology over encrypted traffic: A systematic literature review. *Sensors* 24, 3 (2024), 898.

[7] Leyli Karaçay, Erkay Savaş, and Halit Alptekin. 2020. Intrusion detection over encrypted network data. *Comput. J.* 63, 1 (2020), 604–619.

[8] Nagireddy Karri and Sandeep Kumar Jangam. 2021. Security and Compliance Monitoring. *International Journal of Emerging Trends in Computer Science and Information Technology* 2, 2 (2021), 73–82.

[9] Manh-Dung Nguyen, Anis Bouaziz, Valeria Valdes, Ana Rosa Cavalli, Wissam Mallouli, and Edgardo Montes De Oca. 2023. A deep learning anomaly detection framework with explainability and robustness. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*. 1–7.

[10] Miguel Nicolau, James McDermott, et al. 2018. Learning neural representations for network anomaly detection. *IEEE transactions on cybernetics* 49, 8 (2018), 3074–3087.

[11] Md Mukidur Rahman, Md Shadman Soumik, Md Sheikh Farids, Chowdhury Amin Abdullah, Badhon Sutrudhar, Mohammad Ali, and MD SHAHADAT HOSSAIN. 2024. Explainable anomaly detection in encrypted network traffic using data analytics. *Journal of Computer Science and Technology Studies* 6, 1 (2024), 272–281.

[12] Sadaf Sattar, Shumaila Khan, Muhammad Ismail Khan, Ainur Akhmediyarova, Orken Mamyrbayev, Dinara Kassymova, Dina Oralbekova, and Janna Alimkulova. 2025. Anomaly detection in encrypted network traffic using self-supervised learning. *Scientific Reports* 15, 1 (2025), 26585.

[13] Tao Song, Calvin Ko, Chinyang Henry Tseng, Poornima Balasubramanyam, Anant Chaudhary, and Karl N Levitt. 2005. Formal reasoning about a specification-based intrusion detection for dynamic auto-configuration protocols in ad hoc networks. In *International Workshop on Formal Aspects in Security and Trust*. Springer, 16–33.

[14] Muhammad Usman, Mian Ahmad Jan, Xiangjian He, and Jinjun Chen. 2019. A survey on representation learning efforts in cybersecurity domain. *ACM Computing Surveys (CSUR)* 52, 6 (2019), 1–28.

[15] Fulvio Valenza, Tao Su, Serena Spinoso, Antonio Lioy, Riccardo Sisto, Marco Vallini, et al. 2017. A formal approach for network security policy validation. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.* 8, 1 (2017), 79–100.

[16] John Violos, Fotios Voutsas, Christos Diou, and Aris Leivadeas. 2025. Detecting application transitions and identifying application types for intent-based network assurance: A machine learning perspective. *Computer Networks* (2025), 111872.

[17] Serhii Vladov, Victoria Vysotska, Svitlana Vashchenko, Serhii Bolvinov, Serhii Glubochenko, Andrii Repchonok, Maksym Korniienko, Mariia Nazarkevych, and Ruslan Herasymchuk. 2025. Neural Network IDS/IPS Intrusion Detection and Prevention System with Adaptive Online Training to Improve Corporate Network Cybersecurity, Evidence Recording, and Interaction with Law Enforcement Agencies. *Big Data and Cognitive Computing* 9, 11 (2025), 267.

[18] Jin Yang, Gang Liang, Beibei Li, Guozhu Wen, and Tianyu Gao. 2021. A deep-learning-and reinforcement-learning-based system for encrypted network malicious traffic detection. *Electronics Letters* 57, 9 (2021), 363–365.

[19] Tangda Yu, FuTai Zou, Linsen Li, and Ping Yi. 2019. An encrypted malicious traffic detection system based on neural network. In *2019 international conference on cyber-enabled distributed computing and knowledge discovery (CyberC)*. IEEE, 62–70.

[20] Yi Zeng, Huaxi Gu, Wenting Wei, and Yantao Guo. 2019. $Deep - Full - Range$: a deep learning based network encrypted traffic classification and intrusion detection framework. *IEEE Access* 7 (2019), 45182–45190.

[21] Hao Zhang, William Banick, Danfeng Yao, and Naren Ramakrishnan. 2012. User intention-based traffic dependence analysis for anomaly detection. In *2012 IEEE symposium on security and privacy workshops*. IEEE, 104–112.

[22] Xueqin Zhang, Jiyuan Wang, and Shinan Zhu. 2021. Dual generative adversarial networks based unknown encryption ransomware attack detection. *IEEE Access* 10 (2021), 900–913.

[23] Zhicheng Zhang, Philippe Polet, Frédéric Vanderhaegen, and Patrick Millot. 2004. Artificial neural network for violation analysis. *Reliability Engineering & System Safety* 84, 1 (2004), 3–18.

[24] Ziming Zhao, Zhaoxuan Li, Jialun Jiang, Fengyuan Yu, Fan Zhang, Congyuan Xu, Xinjie Zhao, Rui Zhang, and Shize Guo. 2023. ERNN: Error-resilient RNN for encrypted traffic detection towards network-induced phenomena. *IEEE Transactions on Dependable and Secure Computing* (2023).