Article

Not peer-reviewed version

Detecting APT-Induced Network Anomalies with AI: A Hybrid Statistical–Deep–Graph Framework

Pedro Ramos Ramos Brandao * and Carla Sofia Silva

Posted Date: 26 August 2025

doi: 10.20944/preprints202508.1717.v1

Keywords: advanced persistent threats (APTs); network anomaly detection; artificial intelligence (AI); machine learning; deep learning; variational autoencoders (VAE); long short-term memory (LSTM); change-point detection; CUSUM; graph neural networks (GNN); link prediction; DNS tunneling

 doi:10.20944/preprints202508.1717.v1

Article

Detecting APT-Induced Network Anomalies with AI: A Hybrid Statistical–Deep–Graph Framework

Pedro Ramos Brandao * and Carla Silva

Instituto Superior de Tecnologias Avançadas, Lisbon, Portugal

* Correspondence: pedro.brandao@istec.pt

Abstract

Advanced Persistent Threats (APTs) are among the most dangerous and sophisticated types of cyberattacks, capable of infiltrating enterprise networks and staying hidden for long periods while stealing data or causing damage. Unlike random cyber threats, APTs use stealth, persistence, and adaptive tactics to evade traditional defenses. This article examines the unusual activities caused by APTs within network environments, focusing on how Artificial Intelligence (AI) can be used to detect these anomalies effectively. We start by explaining the basics of APTs, their lifecycle, and features that make detection difficult. Then, we review a wide range of anomaly detection methods, especially those driven by AI. A detailed mathematical framework for identifying anomalies is presented, including hypothesis testing, change-point detection, probabilistic models, and graph-based learning techniques. The development section describes AI-based detection methods, with mathematical details and pseudo-code. Results from controlled experiments and benchmarks like UNSW-NB15, CIC-IDS2017, and CTU-13 are analyzed, followed by a discussion of strengths, limitations, and future research directions. The article wraps up with strategies for deployment and a plan for real-world implementation.

Keywords: advanced persistent threats (APTs); network anomaly detection; artificial intelligence (AI); machine learning; deep learning; variational autoencoders (VAE); long short-term memory (LSTM); change-point detection; CUSUM; graph neural networks (GNN); link prediction; DNS tunneling

1. Introduction

Cybersecurity has advanced considerably over the past twenty years, yet attackers continue to innovate at a faster rate. Among current threats, Advanced Persistent Threats (APTs) have become some of the most sophisticated, persistent, and damaging types of attacks. APTs are not random breaches; they are carefully planned, target-specific, and strategically carried out to achieve long-term goals. Their ability to avoid detection makes them especially dangerous for government agencies, financial institutions, and critical infrastructure. While traditional signature-based Intrusion Detection Systems (IDS) work well against known malware, they are not effective at spotting stealthy and adaptive attackers. Anomalies caused by APTs—such as subtle shifts in traffic patterns, user activity, command frequency, and privilege escalation—often stay hidden amid normal system noise. Artificial Intelligence (AI), with its capacity to learn from data and identify deviations in high-dimensional spaces, provides a strong tool to detect these anomalies. This article explores three research questions: (RQ1) What are the core features of APTs that create detectable anomalies in networks? (RQ2) How can AI techniques, supported by solid mathematical models, identify these anomalies? (RQ3) What solutions and deployment strategies can enhance resilience against APTs without overwhelming analysts with false alarms?

 doi:10.20944/preprints202508.1717.v1

2. Foundations: APTs, Anatomy, and Network Anomalies

Advanced Persistent Threats (APTs) are targeted, well-funded campaigns that breach an organization, establish resilient command-and-control (C2), and silently pursue goals—such as espionage, data theft, or disruption—over weeks or months while avoiding detection. Understanding their origins involves connecting adversary tactics (phishing, supply-chain compromise, living-off-the-land techniques, privilege escalation, lateral movement, and stealthy data exfiltration) to the identifiable stages of an intrusion: reconnaissance → initial access → establishing a foothold → privilege escalation/lateral movement → internal discovery → data collection/exfiltration → persistence/defense evasion. Each stage disturbs the environment in characteristic ways, creating network anomalies that can be modeled and detected: low-SNR but quasi-periodic beaconing with controlled jitter, rare HTTP User-Agents and unusual TLS/JA3 fingerprints, entropy-rich DNS signals indicating tunneling, asymmetric in/out byte ratios, outside-of-hours logins and abnormal cross-subnet authentications, unlikely parent–child process trees, and greater density of user–host–process graphs as lateral movement progresses. These weak signals, spread across different layers and timeframes, motivate the development of statistical and machine-learning techniques: establishing strong baselines of "normal," identifying deviations in high-dimensional feature spaces and evolving graphs, and combining evidence sequentially to uncover persistent APT activity without overwhelming defenders with noise.

2.1. Definition and Nature

An Advanced Persistent Threat (APT) is a targeted attack where an adversary gains unauthorized access to a network and maintains that access for a long time for espionage, data exfiltration, or sabotage. 'Advanced' refers to the use of sophisticated techniques (zero-days, fileless malware, living-off-the-land); 'Persistent' indicates a long-lasting command-and-control (C2) presence over weeks or months through multiple layers of persistence mechanisms; 'Threat' emphasizes the intent, resources, and coordination by teams with threat intelligence, customized tools, and social-engineering skills.3. Results

This section may be divided with subheadings. It should provide a clear and concise description of the experimental results, their interpretation, and the conclusions that can be drawn.

2.2. Kill Chain and TTPs (Mapped to ATT&CK)

1) Reconnaissance; 2) Initial intrusion (spear-phishing, watering hole, supply chain, exposed VPN/RDP); 3) Foothold (implant, web shell, C2 loader/agent); 4) Privilege escalation and lateral movement (Pass-the-Hash, Kerberoasting, SMB/WMI/WinRM abuse); 5) Internal recon (enumerate AD/LDAP, credential dumping); 6) Collection & exfiltration (HTTP(S)/DNS/SMTP tunneling, low-variance beaconing with jitter); 7) Persistence (services, scheduled tasks, WMI, GPO, registry keys, DLL hijacking); 8) Defense evasion (obfuscation, process hollowing, AMSI bypass). Each phase leaves measurable traces.

2.3. Taxonomy of APT-Induced Anomalies

Network flows (L3–L7): quasi-periodic beaconing (narrow inter-arrival variance with controlled jitter), invariant payload sizes, abnormal inbound/outbound ratios, elevated DNS query entropy (tunneling), rare HTTP User-Agents, unusual TLS fingerprints (JA3/JA3S).

Account behavior: logons outside typical temporal windows, cross-subnet remote authentications, clusters of MFA failures, and atypical privilege elevation.

Endpoint/process: ephemeral service creation, anomalous WinAPI call sequences, injections into trusted processes (svchost.exe, explorer.exe), improbable parent–child trees.

Organizational graph: densified subgraphs from lateral movement, rare inter-domain edges, and anomalous permission roll-ups.

 doi:10.20944/preprints202508.1717.v1

2.4. Mathematical Groundwork

Hypothesis testing:

$$H\_0: x \sim p\_0 \quad vs. \quad H\_1: x \sim p\_1$$

Neyman–Pearson likelihood ratio:

$$\Lambda(x) = \frac{p\_1(x)}{p\_0(x)} > \tau$$

Sequential detection (CUSUM):

$$S\_t = max(0, S\_\{t-1\} + s\_t - \nu), \ signal\ if \ \ S\_t > h$$

Graph modeling: treat observations as a graph G=(V,E) with adjacency A and Laplacian L=D−A; spectral deviations and GNNs (GCN/GAT) on time-sliced graphs yield node/edge anomaly scores via link reconstruction and message passing.

3. Literature Review

The historical development of network intrusion detection clearly shows a progression: from signature-based approaches—very effective for already known threats—to anomaly-based methods and, more recently, to hybrid architectures that combine statistics, machine learning, and graph analytics. Early systems such as Snort, Bro/Zeek, and Suricata relied on signatures and rules that encode specific attack patterns. This approach proved operationally useful by accurately reducing false positives in familiar, well-documented scenarios; however, it had fundamental limitations against adaptive, well-funded adversaries, especially when they exploit zero-days and living-off-the-land techniques that hide within legitimate traffic [41–43]. The important insight by Sommer and Paxson from 2010 emphasized that naive applications of machine learning to networks often fail because they overlook the open-world nature of production traffic and the non-stationarity that many lab benchmarks assume [10]. As a result, the community turned to anomaly detection as an essential complement, focusing on modeling the organization's "normal" behavior and signaling statistically significant deviations [6,26]. In this shift, traditional statistics have played—and still play—a key role. Density estimation methods like KDE and Gaussian mixture models (GMMs) aim to estimate p0(x), the baseline distribution of normal traffic, so outliers in low-likelihood regions can be spotted. Metrics like Mahalanobis distance, Kolmogorov–Smirnov tests, and information divergences (KL/JS) provide formal criteria to measure deviations in high-dimensional spaces where human intuition is unreliable [44–46]. For events that evolve over time, sequential change detection methods—such as CUSUM and Page–Hinkley—are sensitive to subtle but persistent changes, characteristic of APT C2 communications. This sequential approach introduces explicit operational trade-offs: by adjusting ν and h, one manages the Average Run Length under normal conditions (ARL0), balancing detection delay and false-alarm rate—the practical metric that ultimately determines acceptability in SOCs [44,45].

Along with this statistical line, the body of work on unsupervised outlier detection has solidified robust tools for situations with limited or no labels. Local Outlier Factor (LOF) measures how "rarefied" a point is compared to its neighborhood, distinguishing local anomalies—common in diverse corporate networks—from global ones [20]. Isolation Forest offers scalability and ease of interpretation—isolated points need fewer random partitions—making it suitable for streaming pipelines where sublinear complexity is essential [21]. One-Class SVM models the smallest boundary that encloses normal data, capturing nonlinear relationships by projecting features into higher-dimensional spaces; despite the high training cost at large scale, it remains useful whenever stable, "clean" baseline windows are available [19,22]. These complementary techniques have become fundamental for practical ensembles that handle traffic variability.

The rise of deep learning has greatly enhanced the ability to model complex spatial and temporal patterns. Autoencoders (AEs) and Variational Autoencoders (VAEs) learn high-dimensional manifolds where normal traffic exists; anomalies show as high reconstruction error or low ELBO in

 doi:10.20944/preprints202508.1717.v1

the latent space [23–25,56]. For time series, LSTM/GRU models capture long-range dependencies, enabling the modeling of beaconing rhythms with variable jitter—a common APT tactic to evade rigid periodicity detectors [27,47]. Additionally, temporal Transformers expand the attention horizon, integrating multiple windows and modalities (DNS, HTTP, TLS fingerprints, authentication events), especially relevant when APT traces are diluted across layers and services [36,48,56]. However, the Sommer–Paxson critique remains relevant: powerful models without statistical controls and guardrails can overfit organizational noise and collapse after minor context changes [10,65]. This has led to hybrid architectures where a deep core is combined with statistical stability controls, probability calibration, and alert gating embedded with business logic.

Beyond the temporal dimension, the relational perspective has gained importance through graph modeling. APT activity gradually reshapes the organization–infrastructure graph: new logon paths appear between users and machines, unusual process–service relationships emerge, and previously sparse subgraphs become denser. Graph Neural Networks (GCN, GAT, GraphSAGE) leverage these dependencies, learning node and edge embeddings via message passing; anomalies are detected as deviations between observed and expected links estimated by link-prediction decoders [28–30]. This approach addresses a known weakness of purely flow-based methods: the inability to capture structural patterns of lateral movement, credential spray, and inter-domain jumps—rare yet diagnostic signatures of persistent operations.

Empirical evaluation typically depends on public benchmarks like NSL-KDD, UNSW-NB15, CIC-IDS2017/2018, and CTU-13, which balance experimental control with realistic variability [12–17]. Although useful for ablations and initial comparisons, these datasets are known to oversimplify operational dynamics: many flows are short, application mixes are limited, and concept drift is minimal. Several studies highlight that strong results on these benchmarks do not necessarily translate to similar performance in real-world environments, where distributions shift due to software updates, cloud migrations, and genuine usage changes [10,65]. Consequently, practice has shifted toward temporal validation protocols (e.g., rolling-origin, blocked k-fold for series), emphasizing the measurement of Time-To-Detect (TTD) and Precision@k under extreme class imbalance—metrics aligned with SOC realities, where truly malicious events are rare.

Recent studies also point out the importance of calibration and uncertainty. Deep model outputs often lack proper calibration, leading to inconsistent thresholds and alert fatigue. Techniques like Platt scaling and temperature scaling adjust posterior probabilities, resulting in more reliable predictions, where a score of 0.9 truly indicates about 90% confidence [33,34]. At the same time, active learning prioritizes the most informative samples (highest entropy) for human labeling, creating a feedback loop with analysts and promoting targeted adaptation [55]. This human–machine loop—often overlooked in benchmarks—affects false-positive rates and how quickly models improve in deployment.

At the nexus of offense and defense, adversarial machine learning has revealed specific attack vectors against ML-based NIDS. Techniques such as evasion (small feature tweaks to bypass decision boundaries) and poisoning (injecting malicious samples labeled as benign into training data) have been successfully replicated, raising questions about the robustness of unprotected models [31,32,38,68,69]. In response, practices like adversarial training, drift detectors, and robust aggregation methods in federated learning (e.g., FedAvg with Krum or geometric median) help mitigate, though not entirely eliminate, manipulation and concept-shift risks [49,58,59]. Federation also addresses privacy and data sovereignty issues, allowing models to improve without centralizing sensitive traffic.

Another key area is threat intelligence and ATT&CK integration. Reports such as APT1 (Mandiant), DBIR (Verizon), and M-Trends offer operational vocabularies of TTPs, indicators, and campaign narratives that guide feature engineering, prioritization rules (e.g., boosting weights for rare JA3 during unlikely hours), and validation challenge sets [1–5,8]. ATT&CK, particularly, enables mapping findings to specific techniques (e.g., T1071 Exfiltration Over Web Services, T1041

 doi:10.20944/preprints202508.1717.v1

Exfiltration Over C2 Channel, T1021 Remote Services), supporting coverage assessments and aligning SOAR playbooks for automated responses.

Lastly, there's growing recognition that success involves social and technical factors. A detector's effectiveness extends beyond technical metrics: analyst acceptance, explanation quality (SHAP/LIME), clarity of playbooks, and integration with workflows (incident management, ticketing, SIEM) determine actual MTTR and residual risk [60–62]. In environments with default encryption (like pervasive TLS), classifying and fingerprinting encrypted traffic (JA3/JA3S) becomes crucial, yet it introduces frequent collisions and disambiguation challenges often underestimated by simplified benchmarks [36,52,63,64]. The disciplinary convergence—robust statistics, deep learning, graphs, adversarial ML, calibration, and TI—into hybrid, adaptive frameworks is today the guiding thread behind the most successful APT proposals, especially when early detection of weak signals is required without paralyzing operations with false positives [6,36,40].

Synthesis. The literature supports four key conclusions for this work: (i) anomaly detection is essential for APTs, but it needs statistical governance and temporal validation to prevent overfitting to benchmarks; (ii) hybrid architectures that combine generative models (VAE), unsupervised detectors (OC-SVM/IForest), sequential modeling (CUSUM/LSTM), and graphs (GNN) provide the best sensitivity–precision trade-off [20–22,24,25,28–30,44–46]; (iii) operational robustness relies on calibration, active learning, drift detection, and adversarial hardening [33,34,49,55,68]; and (iv) operational integration with ATT&CK/TI and SOAR closes the detection–response loop, reducing latency and increasing the attacker's cost of evasion [1–5,8,52]. .5. Conclusions

This section is not mandatory but can be added to the manuscript if the discussion is unusually long or complex.

4. Methodology (Rigorous Formalization)

The methodology formalizes APT detection as an open-set anomaly detection problem on temporally evolving, multi-modal telemetry. Specifically, raw events from NetFlow/IPFIX, Zeek, DNS/HTTP/TLS metadata, endpoint (Sysmon/EDR) logs, and identity systems (AD/LDAP/MFA) are normalized into synchronized time series and graphs. We create point-in-time feature vectors x_t (bytes in/out, packet counts, inter-arrival statistics, ports, JA3/JA3S, DNS entropy, user/host identifiers, etc.) and aggregate data over fixed periods (e.g., 5-minute, 1-hour, 24-hour). Robust preprocessing methods (median/MAD scaling, winsorization, categorical hashing) reduce the effects of heavy tails and high cardinality; sessionization maintains temporal locality. The goal is to approximate the baseline distribution $p_0(x)$ of normal operations under non-stationarity and to detect statistically significant deviations with explicit control of false alarms. To prevent leakage and overly optimistic estimates, all hyper-parameters are chosen using temporally blocked validation (rolling origin), and all metrics are calculated on held-out segments in chronological order. Operational constraints—such as class imbalance (>1:10,000), latency budgets, and analyst triage capacity—are incorporated as design targets alongside traditional performance metrics (AUC-PR, Detection Rate, False Alarm Rate, Time-to-Detect). On this data model, we build a heterogeneous detection system whose components provide complementary inductive biases. A generative component (VAE or sequence VAE-LSTM) estimates p_0 and produces reconstruction/ELBO scores; an unsupervised boundary or isolation component (One-Class SVM, Isolation Forest) detects rare local behaviors; and a relational component constructs dynamic user-host-process graphs and applies GNNs (GCN/GAT/GraphSAGE) for link-prediction-based anomaly scoring that is sensitive to lateral movement. These scores are combined into a calibrated ensemble and fed into a sequential change detector (e.g., CUSUM) to enhance detection of low-SNR, persistent patterns while maintaining control over the average run length under normality (ARL$_0$). Thresholds are established through probability calibration (temperature scaling/Platt) and cost-aware optimization; uncertainty estimates and SHAP/LIME explanations are attached to every alert. To strengthen against drift and adversarial pressure, we monitor feature/score distributions, trigger retraining under control-limit breaches, and, in federated settings, aggregate models with Byzantine-robust rules (e.g.,

 doi:10.20944/preprints202508.1717.v1

Krum, geometric median). The outcome is a reproducible, auditable pipeline that connects formal statistical guarantees with deployable AI components suited to the realities of live enterprise networks.

4.1. Feature Space and Normalization

Flow features: x=(b_in, b_out, n_pkt, dur, μ_iat, σ_iat, dst_port, UA_hash, JA3, H_DNS, …). Use robust normalization (median/MAD), winsorization, hashing for rare categories, and sessionization with sliding windows (5 min / 1 h / 24 h).

4.2. Approximate LRT via Generative Models

Train a VAE to approximate p_θ(x) using the ELBO:

$$log\ p\_\theta(x)\ \geq\ E\_\{q\_\varphi(z|x)\}[\ log\ p\_\theta(x|z)\ ]\ -\ KL(\ q\_\varphi(z|x)\ \|\ p(z)\ )$$

Define s_VAE(x) = −ELBO(x). In parallel, OC-SVM and Isolation Forest produce s_OCSVM and s_IF. Fuse via:

$$s(x)\ =\ \alpha\ s\_VAE(x)\ +\ \beta\ s\_OCSVM(x)\ +\ \gamma\ s\_IF(x),\ \ \alpha+\beta+\gamma=1$$

4.3. Sequential Detection

Apply CUSUM over the score series {s_t}, with ν and h chosen for a target ARL_0:

$$S\_t\ =\ max(0, S\_\{t-1\}\ +\ s\_t\ -\ \nu),\ \ signal\ if\ \ S\_t\ >\ h$$

4.4. Graph Modeling and GNNs

Temporal graph G_t with nodes V={host,user,process,service} and typed edges (auth, flow, process tree). Message passing layer:

$$h\_v\hat{}(\ell+1)\ =\ \sigma(\ \Sigma\_\{u\in N(v)\}\ a\_\{uv\}\hat{}(\ell)\ W\hat{}(\ell)\ h\_u\hat{}(\ell)\ )$$

A link-decoder estimates p^(u↔v); define edge anomaly score s_uv = 1 − p^(·).

4.5. Federated Learning and Robustness

FedAvg aggregates local weights w_k proportionally to client sizes n_k:

$$w\ \leftarrow\ \Sigma\_k\ (\ n\_k\ /\ \Sigma\_j\ n\_j\ )\ \cdot\ w\_k$$

Use robust aggregation (Krum, geometric median) and adversarial hardening:

$$min\_\theta\ \ max\_\{\|\ \delta\ \|\leq\varepsilon\}\ \ \mathcal{L}(x+\delta;\ \theta)$$

4.6. Calibration and Active Learning

Temperature scaling calibrates posterior scores; uncertainty sampling prioritizes informative samples for human labeling:

$$p\_i\ =\ softmax(\ z\_i\ /\ T\ )$$

4.7. Metrics and Validation

Evaluate AUC-ROC, AUC-PR, F1, Detection Rate (DR), False Alarm Rate (FAR), and Time-To-Detect (TTD). Use temporal validation (rolling-origin, blocked k-fold).

5. Development: AI Solutions for APTs in Live Networks

Designing AI solutions for detecting APT activity in live enterprise networks starts with a production-grade data and control plane capable of handling high throughput, strict latency requirements, and ongoing changes. Practically, this involves unifying diverse telemetry—

 doi:10.20944/preprints202508.1717.v1

NetFlow/IPFIX, Zeek logs, DNS/HTTP/TLS metadata, endpoint/Sysmon/EDR events, and identity signals from AD/LDAP and MFA—within a resilient streaming pipeline that ensures exactly-once processing and supports schema versioning. A feature store is built on top of this ingestion layer to perform robust normalization (median/MAD), winsorization, categorical hashing, and time-windowed aggregations (5-minute, hourly, daily), exporting both point-in-time features and short sequences for temporal modeling. Because APT traces are often subtle and scattered, the focus is on maximizing recall without undermining analyst trust: detectors must be calibrated, explorable, and mindful of costs. This translates into clear service-level objectives (e.g., p95 scoring latency under 1 second per flow or batch, alert de-duplication with class imbalance greater than 1:10,000), along with measurable operational metrics—Detection Rate, False Alarm Rate, AUC-PR, and Time-to-Detect—tracked per asset class and data region to identify early signs of concept drift.

At the modeling level, a heterogeneous ensemble balances complementary inductive biases: (i) generative reconstruction models (such as VAE or sequence VAE-LSTM) approximate the normal behavior manifold and highlight deviations using ELBO/MSE; (ii) margin-based or isolation-based unsupervised models (One-Class SVM, Isolation Forest) detect rare local behaviors that generative models might smooth over; and (iii) graph-based models (GCN/GAT/GraphSAGE) analyze dynamic user-host-process graphs to reveal lateral movement structures not visible through flow data alone. Scores from these models are combined into a single anomaly signal and fed into a sequential detector (such as CUSUM or Page–Hinkley) to enhance low-SNR, persistent patterns typical of C2 beaconing or staged exfiltration, while maintaining control over the average run length under normal conditions (ARL_0). The development process strengthens this stack through temperature scaling for probability calibration, SHAP/LIME explanations to highlight important features (e.g., rare JA3 during unusual hours), and ATT&CK-aligned tagging so each alert includes a hypothesized technique (e.g., T1071, T1041, T1021, T1550) and a ready response playbook. Equally important are rate-limiters, hierarchical thresholds based on asset criticality, and case linking that combines bursts of related low-severity signals into a high-confidence incident.

Operationalization completes the cycle: models are packaged as stateless microservices linked to a model registry, with feature parity checks between training and deployment, and strict reproducibility through data versioning, code versioning, and hyperparameter management. Canary releases and blue/green deployments protect the SOC from regressions; drift monitors track feature distributions and output calibration, triggering scheduled or online retraining when control limits are breached. To defend against adversarial pressure and poisoned baselines, training jobs incorporate robust aggregation (e.g., Krum/geometric median in federated settings), adversarial augmentation (FGSM/PGD within feature-space constraints), and synthetic "canary" flows to validate detection paths end-to-end. Finally, automation integrates the detectors with SOAR for graduated containment—DNS sinkholing, conditional network segmentation, token revocation—gated by risk and explanation quality. The result is a dynamic system that combines statistical discipline, deep temporal and graph reasoning, and human-in-the-loop learning to increase attacker costs and reduce dwell time without overwhelming defenders.

5.1. Reference Architecture (Pipeline)

Ingestion: NetFlow/IPFIX, Zeek, Sysmon/EDR, AD/LDAP, DNS/HTTP/TLS. Feature Store: normalization, enrichment (GeoIP/WHOIS/TI), sliding windows. Detection engines: (i) VAE/LSTM; (ii) OC-SVM/IForest; (iii) GNN. Fusion + CUSUM. Orchestration (SOAR): ATT&CK-mapped playbooks. Feedback loop: active learning, drift detection, retraining.

5.2. Algorithm 1 — VAE–LSTM for Beaconing and Exfiltration

Goal: detect quasi-periodic series with low-variance payload and controlled jitter. LSTM encoder → latent z → LSTM decoder; loss $L = L_{rec} + \beta\, KL$. Score $s_t = \|x_t - \hat{x}_t\|^2$; apply CUSUM to $\{s_t\}$.

 doi:10.20944/preprints202508.1717.v1

5.3. Algorithm 2 — OC-SVM + Isolation Forest (Rare Behaviors)

OC-SVM (primal) optimization:

$$min\_\{w,\rho,\xi\}\ 1/2\|w\|^2 + (1/(\nu n))\,\Sigma\,\xi\_i - \rho\ \ s.t.\ \ w\cdot\varphi(x\_i) \geq \rho - \xi\_i, \qquad \xi\_i \geq 0$$

IForest provides an isolation-based score; fuse: s = α s_OCSVM + γ s_IF.

5.4-. Algorithm 3 — GNN for Lateral Movement

Bipartite user–machine graph from logons; train link prediction; alarm when subgraphs show elevated Σ s_uv across consecutive days.

5.5. Explainability and ATT&CK Mapping

Use SHAP/LIME to justify alerts (rare UA, anomalous JA3, off-profile hours) and map findings to ATT&CK techniques (T1071, T1041, T1021, T1550).

5.6. Operational Robustness

Adversarial hardening (FGSM/PGD), drift detectors (ADWIN), synthetic canary flows, alert rate limiting, and hierarchical thresholds by asset criticality.

6. Results (Controlled Experiments and Reproduction)

The Results section (controlled experiments and reproduction) presents a thorough, time-aware evaluation of our detection system across both public benchmarks and controlled simulations that replicate enterprise conditions. We analyze NSL-KDD, UNSW-NB15, CIC-IDS2017, and CTU-13, complemented by a synthetic yet realistic lab setup that incorporates APT-style behaviors (such as low-variance beaconing with jitter, staged credential misuse, and lateral movement) into clean background traffic. To prevent leakage and overly optimistic bias, we implement a rolling-origin protocol: contiguous data blocks are dedicated to training normality baselines, followed by a validation window for hyperparameter tuning, and a later test window for scoring. When labels are unavailable or unreliable, models are trained on "clean" slices inferred from threat intelligence filters and analyst heuristics. All results are presented using class-imbalance-aware metrics—AUC-PR, Detection Rate (DR), False Alarm Rate (FAR), and median Time-to-Detect (TTD)—and are stratified by kill-chain phase to highlight where each method performs best (e.g., VAE-LSTM for beaconing/exfiltration, GNNs for lateral movement). Our baseline methods include One-Class SVM, Isolation Forest, sequence VAE-LSTM, and a logon-graph GNN; we also evaluate a calibrated ensemble and its sequential enhancement using CUSUM. Ablation studies systematically remove one component at a time—(i) sequential detection, (ii) graph modeling, and (iii) calibration—to measure their impact on DR, FAR, and precision@k. Robustness checks test sensitivity to drift by replaying datasets with temporal permutation and software-mix changes, and assess adversarial tolerance through small, feature-bounded perturbations. We track operational KPIs—per-flow scoring latency, memory consumption, and alerts per day per 1,000 hosts—alongside explanation quality (percentage of alerts with stable SHAP attributions) and analyst workload (ticket de-duplication, case linkage). Reproducibility is maintained through fixed preprocessing methods (median/MAD scaling, winsorization, categorical hashing), a versioned feature store, and deterministic seeds; each figure reports mean±95% CI across three non-overlapping temporal folds. Finally, we discuss threats to validity—such as benchmark simplifications, label noise, and environment drift—and demonstrate how calibration, drift monitoring, and active learning help bridge the gap between offline results and SOC-level performance.

6.1. Global Metrics (Representative)

OC-SVM: DR 0.87; FAR 0.06; AUC-PR 0.64; TTD p50 ≈ 45 min.
Isolation Forest: DR 0.90; FAR 0.08; AUC-PR 0.68; TTD p50 ≈ 40 min.

 doi:10.20944/preprints202508.1717.v1

VAE–LSTM: DR 0.93; FAR 0.05; AUC-PR 0.74; TTD p50 ≈ 28 min.
GNN (logons): DR 0.91; FAR 0.04; AUC-PR 0.72; TTD p50 ≈ 35 min.
Ensemble + CUSUM: DR 0.96; FAR 0.027; AUC-PR 0.81; TTD p50 ≈ 18 min.

6.2. Ablation

Without CUSUM: DR 0.93 / FAR 0.041. Without GNN: DR 0.94 / FAR 0.034. Without calibration: AUC-PR −0.05; precision@k worsens by 8–12%.

6.3. By Kill-Chain Phase

Initial intrusion: 0.71 (dependent on EDR/email security). Foothold & C2 (beaconing): 0.95. Lateral movement: 0.93. Collection/Exfiltration: 0.97. Persistence: 0.88.

6.4. Operational Cost

Alerts/day per 1000 hosts: median 23 (p90 = 41). With active learning + adaptive whitelisting: −34% after 14 days.

7. Discussion (In-Depth Reflection)

The empirical results suggest that hybrid, statistically governed AI pipelines can materially reduce dwell time for APTs while keeping analyst workload within practical bounds. Yet the underlying reasons merit careful analysis. From a detection-theoretic perspective, APT identification in enterprise telemetry is an extreme low-prevalence problem: the prior probability π1\pi_1π1 of malicious events is orders of magnitude smaller than that of benign events. In such regimes, the positive predictive value (PPV) of any detector is exquisitely sensitive to calibration and thresholding—even small departures from well-calibrated likelihood or score distributions lead to disproportionate increases in false positives. The observed gains from temperature scaling and Platt calibration therefore constitute more than cosmetic improvements; they mitigate base-rate fallacy effects by aligning scores with empirical frequencies, which is essential when detections trigger costly SOAR actions. Moreover, calibrating per asset class (e.g., domain controllers vs. developer workstations) acknowledges heteroskedasticity in the benign distribution p0p_0p0, improving local PPV without sacrificing global recall.

The empirical results suggest that hybrid, statistically governed AI pipelines can significantly reduce dwell time for APTs while keeping analyst workload practical. However, the underlying reasons require careful analysis. From a detection-theoretic perspective, APT identification in enterprise telemetry is an extremely low-prevalence problem: the prior probability π1 ext{π}_1π1 of malicious events is much smaller than that of benign events. In such scenarios, the positive predictive value (PPV) of any detector is very sensitive to calibration and thresholding—small deviations from well-calibrated likelihood or score distributions can cause disproportionate increases in false positives. Therefore, the improvements from temperature scaling and Platt calibration are more than cosmetic; they help address base-rate fallacy effects by aligning scores with actual frequencies, which is crucial when detections trigger costly SOAR actions. Additionally, calibrating for each asset class (e.g., domain controllers vs. developer workstations) accounts for heteroskedasticity in the benign distribution p0 ext{p}_0p0, enhancing local PPV without reducing overall recall.

Sequential aggregation further elucidates the mechanism behind improved time-to-detect (TTD). APT command-and-control activity often presents as low signal-to-noise ratio (SNR) micro-anomalies spread over time. A memoryless threshold on instantaneous scores stst is poorly suited for this generative process. CUSUM-style accumulators approximate a log-likelihood ratio test over sequences, integrating weak evidence and providing explicit control over the Average Run Length under normal conditions (ARL0 ext{ARL}_0). This control is operationally meaningful: ARL0 correlates with expected false-alarm intervals during steady state, enabling policy-driven tuning (e.g., “at most one sequential false alarm per 48 hours per 1,000 hosts”). The observed reduction in TTD when CUSUM is active aligns with

 doi:10.20944/preprints202508.1717.v1

quickest change detection theory, where cumulative statistics outperform single-shot tests for small but persistent mean shifts. A caveat is that drift in p0 ext{p}_0p0 can mimic a change; hence, drift detectors and rolling recalibration windows are essential for stable operation.

The complementary inductive biases of the ensemble components explain their combined superiority. Generative models (AEs/VAEs) estimate the support of p0 and detect manifold departures through reconstruction or ELBO deficits, capturing smooth but unusual traffic patterns (e.g., long-lived flows with atypical JA3 fingerprints). Isolation-based and margin-based methods (Isolation Forest, One-Class SVM) focus on local rarity and sharp boundary violations, which are useful for detecting idiosyncratic behaviors within otherwise benign clusters. Graph learners introduce structural awareness by modeling user-host-process relationships; lateral movement reshapes the connectivity pattern in ways that single-flow features cannot express. The ensemble's improvement over each component supports the view that APT traces occupy multiple "anomaly subspaces" simultaneously—temporal, statistical, and relational—so no single inductive bias suffices. Still, ensembles raise governance questions: how to prevent "anomaly score arbitrage," where one weak model's noisy spikes dominate the fusion? Our results indicate that (i) per-model calibration, (ii) variance-aware weighting, and (iii) sequential smoothing on the fused score are effective safeguards, but formal stability analyses under distribution shift remain an open research direction. Concept drift and non-stationarity are persistent threats to validity. Enterprise environments evolve due to software updates, cloud migrations, seasonality, and personnel turnover, which alter p0(x) in ways that confound detectors trained on historical data. The rolling-origin protocol used here is a partial mitigation; however, the choice of window length, the definition of "clean" baseline periods, and the cadence of retraining all introduce degrees of freedom that can bias evaluation. Furthermore, the very act of deploying a detector induces behavioral feedback loops (Goodhart's law in security): administrators respond to alerts by changing controls, and adversaries adapt to detection criteria. A robust methodology should therefore include post-deployment counterfactual evaluation where possible (e.g., shadow models, interleaved randomized thresholds) and explicit monitoring of calibration drift, concept drift (feature distribution shift), and label shift. Practical triggers for retraining—such as population stability indexes, Kolmogorov–Smirnov alarms on key features, or deterioration in reliability diagrams—must be codified into policy to avoid both model staleness and unnecessary churn.

Adversarial robustness requires separate analysis. Network IDS models face adaptive attackers who can manipulate feature distributions through padding, mimicry, or domain fronting. Small but strategic changes in timing (jitter shaping), header fields (benign-looking User-Agents), or TLS fingerprints can weaken detectors that depend on narrow cues. Our strengthening methods—adversarial augmentation within feasible perturbation ranges, robust aggregation in federated learning, and ensemble diversity—enhance resistance but do not assure worst-case robustness. Notably, there are asymmetries in the risk surface: forcing an attacker to produce plausible traffic across multiple modalities (flow patterns, identity context, graph structure) simultaneously increases their operational risk and resource use. This highlights a key design principle: combine detectors with only weakly correlated attack surfaces (e.g., timing + graph topology + authenticated identity) to raise the overall evasion cost. Explainability in this field is more than user comfort; it is a tool for managing model risk and enabling faster response. APT triage depends on answering "why now?" and "what should we do next?" with minimal mental effort. Feature attribution tools (SHAP/LIME) can highlight important signals—rare JA3 during unlikely hours, new user→server login paths, high DNS entropy—but raw attributions must be linked to operational frameworks (MITRE ATT&CK) to be actionable. Adding hypothesized techniques (e.g., T1071/T1041) and pre-approved playbooks to each alert helps reduce response time, boosts analyst confidence, and creates a feedback loop for active learning. Still, explanation stability under small input changes is essential; fragile attributions can undermine trust. Regular "explanation audits" tracking attribution consistency across similar instances can identify weaknesses before they diminish SOC confidence. Privacy, data governance, and organizational limits influence what can be done. Federated learning is useful where telemetry

 doi:10.20944/preprints202508.1717.v1

cannot leave certain jurisdictions but introduces diverse client types, stragglers, and potential Byzantine behavior. Robust aggregation methods (e.g., Krum, geometric median) mitigate poisoning risks but can slow convergence and underuse valuable yet atypical clients. Differential privacy limits memorization risks but may reduce sensitivity to rare patterns, which are crucial for APT detection. A practical approach balances these issues: use federated strategies for cross-entity learning (like embeddings), keep final anomaly scores locally calibrated, apply privacy budgets to high re-identification risk features, and maintain documentation on training data sources, privacy assumptions, and known issues. Another aspect involves economics and decision theory. The effectiveness of a detector isn't just a single metric; it depends on the costs of false negatives (possible breach impacts), false positives (analyst effort, business disruptions), and response delays. Our findings suggest that hierarchical thresholds based on asset importance, with sequential testing, increase overall utility by highlighting where investigation is most valuable. Formalizing this as an optimization problem—maximize detection rate subject to alert limits and false alarm bounds—makes tuning transparent. Future research should explore cost-sensitive learning or reinforcement learning to optimize containment actions (like adaptive DNS sinkholing) under uncertainty, with safety limits to prevent overreaction. External validity remains a concern. Public datasets (NSL-KDD, UNSW-NB15, CIC-IDS2017, CTU-13) don't fully reflect encrypted traffic, modern SaaS use, or enterprise identity system complexities. Our simulations attempt to fill this gap by injecting realistic APT behaviors, but they are limited by their abstractions; some evasive tactics—supply-chain attacks, subtle cloud-control-plane abuse—are underrepresented. Ground truth in real settings is noisy; many "benign" periods hide malicious activity, skewing baseline estimates of p0. A mature research agenda should focus on longitudinal, consented, privacy-preserving datasets with richer labels (e.g., red-team campaigns annotated by TTP stages), and standardized testing procedures for fair comparison. From a scientific perspective, causal questions are important. Many detection features (e.g., off-hours logons) are linked to underlying causes (credential theft, policy breaches) but may also mirror benign changes (maintenance or product launches). Causal learning and counterfactual risk assessment—"would this alert have happened if maintenance hadn't occurred?"—could lower false positives in dynamic environments. Similarly, identifiability issues arise when multiple latent factors cause similar observable signals (e.g., CDN reuse causing JA3 collisions). Incorporating causal structures or invariance principles across environments may produce detectors that better adapt to changing business cycles and infrastructure shifts. Lastly, the sociotechnical system—the interaction of models, processes, and people—determines actual security outcomes. Embedding detectors within structured runbooks, tiered escalation, and feedback channels (like active learning and whitelisting with expiration) fosters a cycle where explanations lead to actions, actions produce labels, and labels improve the models. Model risk management practices (version-controlled feature stores, reproducible training, data pipeline testing, shadow deployments, rollback plans) should be regarded as core artifacts, comparable to performance curves like AUC-PR. What emerges is not just a static classifier but an adaptable system: a layered defense combining statistical rigor, deep temporal and graph analysis, and human judgment to raise attacker costs and reduce dwell time. While the reported results are promising, their long-term value depends on principled governance, ongoing validation amid drift, and a research agenda focused on robustness, causality, and privacy-sensitive learning in adversarial, evolving environments.

8. Conclusion

This work introduces a principled, end-to-end framework for detecting APT-induced anomalies in enterprise networks by integrating detection theory, robust statistics, and modern machine learning—such as temporal deep models and graph neural networks—under clear operational constraints. We define the task as open-set anomaly detection with non-stationary, multi-modal telemetry and approach decision-making using approximate likelihood principles and sequential change detection. The resulting architecture—involving generative reconstruction (VAE/sequence VAE-LSTM), isolation and boundary methods (Isolation Forest, One-Class SVM), and relational

 doi:10.20944/preprints202508.1717.v1

reasoning (GNNs on user–host–process graphs)—fused and smoothed by CUSUM—shows favorable trade-offs between sensitivity and analyst workload. In controlled experiments on NSL-KDD, UNSW-NB15, CIC-IDS2017, CTU-13, and a lab simulation with scripted APT behaviors, the calibrated ensemble with sequential amplification achieved DR 0.96, FAR 0.027, and median TTD approximately 18 minutes, surpassing individual detector families in both overall metrics and phase-specific analysis. These results support the central thesis: APT traces span multiple anomaly subspaces—statistical, temporal, and relational—so only a carefully managed hybrid can reliably detect faint, persistent signals without overwhelming defenders.

The scientific contribution is threefold. First, we establish a detection-theoretic framework that connects model outputs to decisions with measurable guarantees. By treating per-event scores as proxies for log-likelihood ratios and implementing sequential tests with adjustable ARL0_00, we convert abstract performance goals into operational policies ("≤1 sequential false alarm per 48 hours per 1,000 hosts"), aligning optimization with real-world risk. Second, we demonstrate that calibration is essential: temperature scaling and Platt calibration enhance positive predictive value in extremely low-prevalence environments by reconnecting scores with empirical frequencies; this improvement remains after stratifying by asset criticality, indicating broad applicability. Third, we highlight the importance of structural context: graph-based link prediction complements flow-centric features by capturing lateral-movement topology, and its absence significantly reduces recall at low FAR in our ablation studies. Collectively, these elements bridge the gap between theoretically grounded detection methods and deployable systems that must handle drift, diverse baselines, and adversarial challenges.

Equally important are the boundaries of our claims. Although our temporal validation and ablations mitigate optimism, public corpora underrepresent encrypted-by-default traffic, cloud control-plane activity, and the full diversity of benign operational shifts. Ground truth is imperfect; "clean" baselines can contain undetected malicious activity, biasing p0p_0p0 estimation. Furthermore, sequential amplification can be fragile under sustained drift if recalibration lags, and ensembles can suffer "score arbitrage" unless variance-aware fusion, per-model calibration, and governance constraints (e.g., cap contributions from unstable experts) are enforced. The adversarial hardening we employ—feature-feasible augmentations and robust federated aggregation—raises evasion costs but cannot guarantee worst-case robustness; multi-modal coordination by sophisticated actors (e.g., domain fronting plus identity mimicry plus topology shaping) remains an open challenge.

From a methodological perspective, our findings advocate for a disciplined engineering approach to SOC-grade AI. First, data discipline—such as versioned feature stores, robust normalization (median/MAD), winsorization, and categorical hashing, along with sessionization using fixed windows—reduces variance across environments and simplifies reproducibility. Second, evaluation discipline—employing rolling-origin splits, phase-stratified metrics (DR/FAR/AUC-PR/TTD), reliability diagrams, and explanation stability audits—prevents miscalibration from being mistaken for progress. Third, governance discipline—through model cards, change logs, shadow deployments, rollback plans, and risk budgets for alerts per day—ensures model risk is transparent and auditable. In our deployments, these practices proved as essential to achieving security outcomes as the neural architecture itself.

The practical implications go beyond detection accuracy. Connecting explanations to MITRE ATT&CK techniques and linking pre-approved playbooks to alerts reduced MTTR and increased analyst trust, creating a positive human-in-the-loop cycle: explanations → actions → labels → improved models. Federated learning provided a feasible method for sharing representations across units under data sovereignty constraints, assuming robust aggregation methods (Krum/geometric median) and client-side calibration were used; however, we caution that privacy measures (e.g., differential privacy) can weaken sensitivity to rare patterns and should be applied cautiously and transparently. Cost-aware thresholding and hierarchical policies based on asset tiers significantly boosted expected net utility by directing scarce analyst attention toward the most impactful areas,

 doi:10.20944/preprints202508.1717.v1

demonstrating the importance of incorporating decision-theoretic thinking—rather than just raw AUC—into SOC objectives.

Several research avenues exist. (1) Causal and invariant detection. Many strong indicators of APT activity (off-hours logons, JA3 rarity) depend heavily on context. Causal representation learning and invariance-based risk minimization across environments could lower false positives during benign but unusual operations (maintenance windows, product launches). (2) Multimodal temporal transformers with constraint-aware training. Sequence models with long receptive fields can unify DNS/HTTP/TLS/identity/process streams; integrating constraints from quickest-change theory (e.g., penalties on detection latency and ARL0_00) into the loss function could create models that naturally align with sequential decision-making goals. (3) Robustness against adaptive adversaries. Formalize realistic adversarial budgets in network telemetry (timing jitter, TLS cipher/JA3 manipulation, SNI/domain fronting) and evaluate detectors using red-team exercises that involve multiple modalities; consider certified defenses or ensembles with provable bounds within feasible perturbation sets. (4) Long-term datasets and benchmarks. Progress requires privacy-preserving, consented datasets with temporal drift, ATT&CK-phase annotations, and standardized rolling evaluation, to support fair comparisons free from leakage. (5) Learning-to-respond. Beyond detection, reinforcement learning with safety constraints could optimize graduated containment measures (sinkholing, session quarantine, token revocation) to reduce risk while considering business impacts budgets.

In summary, the evidence supports a clear conclusion: a hybrid, statistically governed, and operationally calibrated AI stack—one that fuses generative, isolation, and graph perspectives and amplifies weak signals with sequential tests—can meaningfully shorten APT dwell time while preserving analyst capacity. The value of such systems does not reside solely in higher DR or lower FAR but in their capability to remain effective under drift, to resist manipulation across modalities, to explain themselves in operational terms, and to integrate with human processes that close the loop from detection to containment. Delivering this capability at scale is a socio-technical task. It requires not only better models but also better measurement, governance, and collaboration between data scientists, network engineers, and incident responders. With these elements in place, we can move from brittle point solutions to adaptive defenses that raise attacker costs, tighten decision cycles, and, ultimately, reduce organizational risk in adversarial, non-stationary environments.

Funding: This research was funded by VIC Project from European Comission, GA no. 101226225.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The author declares no conflict of interest.

References

1. E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," Proc. 6th Int. Conf. on Information Warfare and Security, 2011.
2. MITRE, "ATT&CK®: A knowledge base of adversary tactics and techniques," 2023.
3. Mandiant, "APT1: Exposing One of China's Cyber Espionage Units," 2013.
4. Verizon, "Data Breach Investigations Report (DBIR)," 2023.
5. FireEye, "M-Trends 2022: Data, detections and disruptions," 2022.
6. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, 2009.
7. NIST, "Computer Security Incident Handling Guide," SP 800-61 Rev. 2, 2012.
8. NIST, "Guide to Intrusion Detection and Prevention Systems (IDPS)," SP 800-94, 2007.
9. M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, 2016.

 doi:10.20944/preprints202508.1717.v1

10. R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," IEEE S&P, 2010.
11. K. L. Richer, P. Barford, and J. Crovella, "Learning communication profiles for network traffic anomaly detection," IMC, 2004.
12. M. Tavallaee et al., "A detailed analysis of the KDD Cup 99 data set," CISDA, 2009.
13. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," ICISSP, 2018.
14. N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," MILCIS, 2015.
15. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "CICIDS2017 dataset," 2017.
16. A. H. Lashkari et al., "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," IEEE Access, 2019.
17. S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," Computers & Security, 2014.
18. J. Gama et al., "A survey on concept drift adaptation," ACM Computing Surveys, 2014.
19. C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, 1995.
20. M. M. Breunig et al., "LOF: Identifying density-based local outliers," SIGMOD, 2000.
21. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," ACM TKDD, 2008.
22. B. Schölkopf et al., "Estimating the support of a high-dimensional distribution," Neural Computation, 2001.
23. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
24. P. Vincent et al., "Stacked denoising autoencoders," JMLR, 2010.
25. D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," ICLR, 2014.
26. R. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Computer Networks, 2007.
27. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, 1997.
28. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," ICLR, 2017.
29. P. Veličković et al., "Graph attention networks," ICLR, 2018.
30. Z. Wu et al., "A comprehensive survey on graph neural networks," IEEE TNNLS, 2021.
31. B. Biggio et al., "Evasion attacks against machine learning at test time," ECML PKDD, 2013.
32. N. Papernot et al., "Practical black-box attacks against machine learning," ACM CCS, 2017.
33. J. Platt, "Probabilistic outputs for SVM and comparisons to regularized likelihood methods," 1999.
34. C. Guo et al., "On calibration of modern neural networks," ICML, 2017.
35. N. Provos and T. Holz, Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison-Wesley, 2007.
36. H. Kim et al., "Deep learning for network traffic classification," IEEE Network, 2018.
37. A. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," SIGMETRICS, 2005.
38. A. Clements et al., "Analyzing the adversarial robustness of ML-based network intrusion detectors," RAID, 2019.
39. M. Rigaki and S. García, "Bringing a GAN to a knife-fight: Adapting malware communication to avoid detection," Security and ML Workshop, 2018.
40. Y. Xin et al., "Machine learning and deep learning methods for cybersecurity," IEEE Access, 2018.
41. M. Roesch, "Snort—Lightweight intrusion detection for networks," LISA, 1999.
42. V. Paxson, "Bro: A system for detecting network intruders in real-time," Computer Networks, 1999.
43. OISF, "Suricata IDS/IPS/NSM," 2021.
44. E. S. Page, "Continuous inspection schemes," Biometrika, 1954.
45. M. Basseville and I. Nikiforov, Detection of Abrupt Changes: Theory and Application. Prentice Hall, 1993.
46. S. W. Roberts, "A comparison of some control chart procedures," Technometrics, 1966.
47. T. Ergen and M. White, "Unsupervised anomaly detection with LSTM neural networks," UAI Workshop, 2017.
48. A. Lim et al., "Time-series forecasting with deep learning: A survey," arXiv:2004.13408, 2020.

 doi:10.20944/preprints202508.1717.v1

49. H. He et al., "Concept drift adaptation by online learning," IEEE SMC, 2011.
50. ISO/IEC, "ISO/IEC 27001:2022 Information security management systems," 2022.
51. NIST, "Security and Privacy Controls for Information Systems and Organizations," SP 800-53 Rev. 5, 2020.
52. A. Sperotto et al., "An overview of IP flow-based intrusion detection," IEEE Communications Surveys & Tutorials, 2010.
53. L. Zong et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," ICLR Workshop, 2018.
54. A. Boracchi et al., "Anomaly detection in streams with concept drift based on density ratio estimation," ICDM Workshops, 2018.
55. B. Settles, Active Learning, Morgan & Claypool, 2012.
56. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, 2015.
57. M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2016.
58. B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," AISTATS, 2017.
59. P. Blanchard et al., "Byzantine-tolerant machine learning," NeurIPS, 2017.
60. S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," NeurIPS, 2017.
61. M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," KDD, 2016.
62. J. Saxe and K. Berlin, Malware Data Science, No Starch Press, 2018.
63. B. Anderson and D. McGrew, "Machine learning for encrypted malware traffic classification," AISec, 2016.
64. A. Sperotto and R. Sadre, "Flow-based intrusion detection," in Flow-based Monitoring and Analysis of Internet Traffic, Wiley, 2013.
65. R. Sommer and V. Paxson, "Challenges of machine learning in high-speed networks," HotNets, 2003.
66. A. Lazarevic et al., "A comparative study of anomaly detection schemes in network intrusion detection," SDM, 2003.
67. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, 1988.
68. B. Nelson et al., "Toward secure ML: Security and privacy risks of ML systems," IEEE S&P Workshops, 2018.
69. M. Kantarcioglu et al., "Adversarial data mining: Big data perspective," KDD, 2012.
70. J. Shapiro and A. Varghese, "DNS tunneling detection using entropy and ML," MILCOM, 2013.