Computer Fraud and Security
ISSN (online): 1873-7056

A Trust-Aware Framework for Detecting Hidden Threat Persistence in Distributed Enterprise Networks

Md Riyad Uddin

Email: raselmit@gmail.com

Master of Science in Computer Science, Westcliff University

Orcid ID: https://orcid.org/0009-0009-2963-668X

Md Anamul Haque

Email: ahaquemb@gmail.com

Master of Science in Information Technology, Washington University of Science and Technology

Orcid id: https://orcid.org/0009-0000-0125-8933

ABSTRACT Advanced Persistent Threats (APTs) increasingly evade traditional signature- and rule-based defenses by mimicking legitimate user behavior over extended dwell periods, often exceeding 200 days in distributed enterprise environments. Existing intrusion detection systems (IDS) and user and entity behavior analytics (UEBA) platforms struggle to differentiate slow, low-and-slow lateral movement from benign operational drift, resulting in high false-negative rates for persistence-stage attacks. This paper proposes a Trust-Aware Hidden Threat Persistence Detection (TA-HTPD) framework that fuses dynamic, graph-based trust scoring with deep behavioral anomaly modeling and MITRE ATT&CK-aligned persistence pattern matching. The framework continuously recalibrates a Bayesian trust score for every network entity based on multi-source telemetry (endpoint, network flow, identity, and cloud audit logs), enabling early detection of trust erosion indicative of credential abuse, lateral movement, and command-and-control beaconing. We formalize the trust update model, derive an adaptive thresholding mechanism, and implement the framework on a hybrid testbed combining the CICIDS2017 dataset with a synthetically augmented lateral-movement corpus emulating a 312-host enterprise topology. Experimental results demonstrate that TA-HTPD achieves a detection precision of 0.93, recall of 0.91, and F1-score of 0.92, outperforming Isolation Forest, standalone LSTM-autoencoder, static rule-based SIEM correlation, and graph-based UEBA baselines by 18.4 to 41.6 percent in F1-score, while reducing mean detection latency for persistent threats from 19.5 hours (static SIEM) to 4.2 hours. These findings indicate that integrating dynamic trust modeling with behavioral and pattern-based detection significantly improves early-stage visibility into hidden threat persistence without proportionally increasing analyst alert volume.

INDEX TERMS *Advanced persistent threat, Bayesian trust model, behavioral anomaly detection, dynamic trust scoring, distributed enterprise networks, intrusion detection, lateral movement, MITRE ATT&CK, trust-aware security, zero trust architecture.*

I. INTRODUCTION

Distributed enterprise networks now span on-premises data centers, multi-cloud workloads, remote endpoints, and third-party integration points, producing an attack surface that is structurally difficult to monitor comprehensively. Within this environment, Advanced Persistent Threats (APTs) represent one of the most damaging categories of intrusion because their objective is not rapid exfiltration but sustained, low-visibility presence that allows long-term reconnaissance, data exfiltration, and disruption capability.

Industry incident-response telemetry consistently reports median adversary dwell times — the interval between initial compromise and detection — in the range of several months for sophisticated intrusions [1], [2]. During this period, attackers perform reconnaissance, escalate privileges, move laterally between hosts, and establish redundant footholds, frequently using legitimate administrative tools and valid credentials (so-called "living-off-the-land" techniques) [3]. Because these actions individually resemble normal administrative activity, conventional signature-based intrusion detection systems (IDS) and static correlation rules in security information and event management (SIEM) platforms generate either an unmanageable volume of false positives or fail to flag the activity at all [4].

User and Entity Behavior Analytics (UEBA) platforms attempt to address this gap by modeling baseline behavior and flagging deviations [5], [6]. However, most UEBA approaches treat each entity independently and apply a static anomaly threshold, which is poorly suited to the gradual, incremental behavioral drift characteristic of persistence-stage APT activity. Moreover, these approaches generally do not account for the relational structure of enterprise networks: a

compromised low-privilege workstation that begins communicating with a domain controller represents a qualitatively different risk than the same anomaly occurring on an isolated test machine, yet behavioral models that ignore topology and trust relationships cannot distinguish between the two.

Zero Trust Architecture (ZTA) principles, formalized in guidance such as NIST SP 800-207 [7], propose continuous verification of every access request rather than implicit trust based on network location. While ZTA reshapes access control, most practical implementations apply trust decisions at the point of authentication and do not maintain a continuously evolving, graph-structured trust state that reflects accumulated behavioral evidence over time. This leaves a gap between perimeter-oriented continuous verification and the longitudinal behavioral analysis required to detect persistence.

This paper addresses that gap by proposing the Trust-Aware Hidden Threat Persistence Detection (TA-HTPD) framework, which unifies three complementary detection signals into a single trust-aware decision process: (i) a Behavioral Drift Estimator (BDE) based on an LSTM-autoencoder that quantifies deviation from per-entity behavioral baselines; (ii) a Dynamic Trust Scoring Module (DTSM) that maintains a Bayesian, graph-propagated trust score for every host, user, and service account, updated continuously as new evidence arrives; and (iii) a Persistence Pattern Matcher (PPM) that correlates flagged sequences against MITRE ATT&CK persistence and lateral-movement techniques [8]. The fusion of these signals through a weighted trust-risk aggregation layer enables the framework to surface low-and-slow attack campaigns substantially earlier than baselines that rely on any single signal in isolation.

The principal contributions of this paper are as follows.

1) We formalize a continuous, graph-propagated Bayesian trust model for heterogeneous enterprise entities (hosts, users, service accounts, and cloud resources), including an adaptive thresholding mechanism that adjusts sensitivity based on local trust-graph topology and historical false-positive rates.
2) We design a multi-signal fusion architecture that combines deep behavioral anomaly detection, dynamic trust scoring, and MITRE ATT&CK-aligned pattern matching, with an explicit feedback loop for adaptive trust re-calibration.
3) We construct a hybrid evaluation testbed combining the CICIDS2017 benchmark dataset [9] with a synthetically generated lateral-movement corpus emulating a 312-host distributed enterprise topology, and we release the topology generation methodology to support reproducibility.
4) We provide a comprehensive empirical evaluation against four representative baselines (Isolation Forest, standalone LSTM-autoencoder, static rule-based SIEM correlation, and graph-based UEBA), reporting precision, recall, F1-score, ROC-AUC, and detection latency, together with an ablation study isolating the contribution of each TA-HTPD component.

The remainder of this paper is organized as follows. Section II reviews related work in APT detection, behavioral analytics, and trust-based security models. Section III presents the system model and formal trust-update equations. Section IV describes the TA-HTPD architecture and its constituent modules. Section V details the experimental methodology, dataset construction, and testbed topology. Section VI presents results and comparative analysis. Section VII discusses limitations, deployment considerations, and threats to validity. Section VIII concludes the paper and outlines directions for future work.

II. RELATED WORK

A. Signature-Based and Anomaly-Based Intrusion Detection

Traditional IDS approaches rely on signatures of known attack patterns, encoded as rules in systems such as Snort and Suricata. While effective against known exploits, these systems are inherently reactive and provide no defense against novel persistence techniques that abuse legitimate credentials and tools [4], [10]. Anomaly-based approaches address this limitation by learning a model of normal behavior and flagging statistical deviations. Classical methods such as Isolation Forest [11] and One-Class SVM [12] have been widely applied to network flow data, achieving reasonable performance on point anomalies such as port scans or volumetric attacks, but performing poorly on the gradual, low-magnitude behavioral drift associated with persistence-stage activity, since each individual deviation may fall within normal variance bounds.

B. Deep Learning for Behavioral Anomaly Detection

Recurrent architectures, particularly LSTM-based autoencoders, have been applied to sequential log and flow data to capture temporal dependencies in user and host behavior [13], [14]. Du et al. [13] proposed DeepLog, which models system log sequences using LSTM networks to detect anomalies in execution patterns. Subsequent work has extended this to network flow sequences [14] and to multivariate time-series of host telemetry [15]. These approaches improve sensitivity to subtle temporal anomalies but typically operate on a per-entity basis without incorporating the relational context of the broader network, and they generally apply a fixed reconstruction-error threshold that does not adapt to the evolving risk posture of the entity being monitored.

C. Graph-Based and UEBA Approaches

Graph-based detection methods represent the enterprise network as a provenance or interaction graph and apply graph neural networks (GNNs) or graph-embedding techniques to identify anomalous substructures consistent with lateral movement [16], [17]. King and Huang [16] demonstrated that provenance graph analysis can recover multi-stage attack campaigns from system audit logs with substantially lower false-positive rates than flow-level analysis alone. UEBA platforms such as those surveyed by Shashanka et al. [18] combine multiple behavioral features (login times, data volumes, access patterns) into composite risk scores, but these scores are typically computed independently per entity and recalibrated on fixed schedules (e.g., daily), limiting their responsiveness to rapidly evolving persistence campaigns.

D. Trust Models and Zero Trust Architecture

Trust management in distributed systems has a long history in peer-to-peer and IoT security research, where Bayesian and reputation-based trust models are used to estimate the reliability of nodes based on observed interactions [19], [20]. NIST SP 800-207 [7] formalizes Zero Trust Architecture principles, emphasizing continuous verification, least-privilege access, and micro-segmentation. However, as noted by Rose et al. [7] and subsequent implementation studies [21], most ZTA deployments apply trust evaluation primarily at access-decision points (e.g., policy enforcement points for authentication) rather than maintaining a continuously updated trust state informed by longitudinal behavioral evidence. Sarkar et al. [22] proposed a dynamic trust scoring model for IoT networks based on weighted historical interaction quality, but this work does not address the specific challenge of mapping trust erosion to known adversary persistence techniques.

E. MITRE ATT&CK-Aligned Detection

The MITRE ATT&CK framework [8] provides a structured taxonomy of adversary tactics, techniques, and procedures (TTPs), including dedicated categories for Persistence (TA0003) and Lateral Movement (TA0008). Several works have proposed mapping detected events to ATT&CK techniques to improve analyst interpretability and enable cross-tool correlation [23], [24]. However, these mapping approaches are generally applied as a post-hoc labeling step over alerts generated by independent detectors, rather than as an integral component of the detection decision itself.

F. Positioning of the Proposed Work

Table 1 summarizes the positioning of TA-HTPD relative to representative prior work along four dimensions: whether the approach models behavioral sequences, whether it maintains a dynamic (continuously updated) trust state, whether it incorporates network-graph topology, and whether detections are aligned to a recognized TTP taxonomy. To the best of our knowledge, no prior framework combines all four properties within a unified, continuously recalibrated decision pipeline, which motivates the design presented in Section IV.

Approach	Behavioral Sequence Modeling	Dynamic Trust State	Graph / Topology Aware	ATT&CK- Aligned Output
Isolation Forest [11]	No	No	No	No
DeepLog (LSTM) [13]	Yes	No	No	No
Provenance-Graph GNN [16]	Partial	No	Yes	No
UEBA Composite Scoring [18]	Partial	Partial (static)	No	No
IoT Bayesian Trust [22]	No	Yes	Partial	No
ATT&CK Post-hoc Mapping [23]	No	No	No	Yes
TA-HTPD (Proposed)	Yes	Yes (continuous)	Yes	Yes

TABLE 1. Comparative Positioning of TA-HTPD Against Representative Prior Approaches

III. SYSTEM MODEL AND PROBLEM FORMALIZATION

A. Network and Entity Representation

We represent the distributed enterprise network as a time-evolving attributed graph G(t) = (V, E(t)), where V is the set of entities — hosts, user accounts, service accounts, and cloud resources — and E(t) is the set of observed interaction edges (authentication events, network connections, file accesses, API calls) active at time t. Each entity v in V is associated with a feature vector x_v(t) in R^d capturing behavioral attributes such as login frequency, process-creation rate, data-transfer volume, and access-pattern entropy, sampled at discrete time steps of length Delta t (set to one hour in our implementation).

B. Behavioral Drift Estimation

For each entity v, the Behavioral Drift Estimator (BDE) maintains a reconstruction model f_theta, implemented as an LSTM-autoencoder, trained on a sliding window of W historical observations during a baseline learning phase. At time t, the BDE produces a reconstruction x-hat_v(t) = f_theta(x_v(t-W:t)) and computes the behavioral deviation as the normalized reconstruction error:

$$d_v(t) = \| x_v(t) - \text{x-hat}_v(t) \|_2 \ / \ \text{sigma}_v \tag{1}$$

where sigma_v is the standard deviation of reconstruction error for entity v observed during the baseline period, used to normalize across entities with heterogeneous activity profiles. A value of d_v(t) close to zero indicates behavior consistent with the learned baseline, while large values indicate significant deviation.

C. Dynamic Bayesian Trust Score

Rather than treating d_v(t) in isolation, the Dynamic Trust Scoring Module (DTSM) maintains a continuous trust score T(v,t) in [0,1] for every entity, interpreted as the posterior probability that the entity is operating within its expected, non-malicious behavioral envelope. The trust score is updated recursively using a Bayesian filter formulation. We model the observation likelihood of the behavioral deviation as a function of the latent trust state, and define the trust update as:

$$T'_v(t) = T_v(t-1) - \text{alpha} * d_v(t) + \text{beta} * (1 - T_v(t-1)) \tag{2}$$

where alpha in (0,1) controls the rate of trust erosion in response to behavioral deviation, and beta in (0,1) is a recovery (regression-to-prior) term that allows trust to gradually return toward neutral in the absence of continued anomalous evidence, preventing permanent penalization from transient noise. T'_v(t) is then propagated across the trust graph to incorporate relational context: an entity that is behaviorally consistent but has recently established new high-frequency connections to low-trust neighbors should also experience trust erosion, since this pattern is characteristic of lateral movement staging. The graph-propagated trust score is computed as:

$$T_v(t) = (1 - \text{gamma}) * T'_v(t) + \text{gamma} * \text{SUM}_{u \text{ in } N(v)} \ w_{uv}(t) * T_u(t-1) \tag{3}$$

where N(v) is the set of neighbors of v in G(t), gamma in [0,1] is a propagation weight controlling the influence of neighbor trust, and w_{uv}(t) is an edge weight proportional to the recency and frequency of interaction between u and v, normalized so that SUM_{u in N(v)} w_{uv}(t) = 1. Equation (3) is the trust update referenced by the flowchart in Fig. 7. Initial trust values T_v(0) are set to 0.9 for all entities following a successful baseline-learning period of two weeks, reflecting a weak prior of legitimacy that can be eroded by subsequent evidence.

D. Adaptive Persistence Threshold

A static global threshold for T_v(t) is inappropriate because acceptable trust ranges vary by entity role (e.g., a domain controller legitimately interacts with many hosts and may exhibit different baseline variance than a single-user workstation). We therefore define an adaptive threshold tau_adaptive(v,t) computed per entity as:

$$\text{tau\_adaptive}(v,t) = \text{mu}_T(v) - k * \text{sigma}_T(v) * (1 + \text{rho} * \text{FPR}_v) \tag{4}$$

where mu_T(v) and sigma_T(v) are the mean and standard deviation of T_v over a trailing 30-day window, k is a global sensitivity constant (set to 2.0 in our experiments, corresponding approximately to a two-sigma deviation), FPR_v is the historical false-positive rate for entity v as estimated from analyst feedback, and rho controls how strongly historical false positives widen the threshold (i.e., reduce sensitivity) for noisy entities. When T_v(t) < tau_adaptive(v,t), the entity is flagged for evaluation by the Persistence Pattern Matcher described in Section IV-C.

E. Multi-Signal Fusion Score

The final persistence risk score R_v(t) used to rank alerts combines the trust deficit, the raw behavioral deviation, and a pattern-match indicator m_v(t) in {0,1} produced by the Persistence Pattern Matcher (Section IV-C), through a weighted aggregation:

$$R\_v(t) = w1 * (1 - T\_v(t)) + w2 * norm(d\_v(t)) + w3 * m\_v(t) \quad (5)$$

where norm(.) denotes min-max normalization over the active entity population at time t, and w1 + w2 + w3 = 1. The weights were tuned via grid search on a validation split (Section V-D) to w1 = 0.45, w2 = 0.25, w3 = 0.30, reflecting the empirical finding that trust-deficit and pattern-match signals are jointly more discriminative than raw behavioral deviation alone. Entities with R_v(t) above a configurable operational threshold are surfaced to analysts in descending order of R_v(t), forming the risk-ranked alert output shown in Fig. 1.

F. Threat Model and Assumptions

We assume an adversary that has obtained an initial foothold (e.g., via phishing or a compromised credential) and seeks to maintain long-term access while avoiding detection through gradual privilege escalation and lateral movement using legitimate protocols (RDP, SMB, WMI, SSH) and valid, though potentially abused, credentials. We assume the monitoring infrastructure itself (log collection pipeline, BDE, DTSM, and PPM) is not compromised, consistent with standard assumptions in the UEBA and SIEM literature [5], [18]. We do not address attacks that occur entirely outside instrumented telemetry sources (e.g., purely physical access) or zero-day exploitation of the monitoring stack itself, which we discuss as limitations in Section VII.

IV. PROPOSED TA-HTPD ARCHITECTURE

The TA-HTPD framework, illustrated in Fig. 1, is organized as a five-layer pipeline: (1) a multi-source data ingestion and normalization layer; (2) three parallel analytic modules — the Behavioral Drift Estimator (BDE), Dynamic Trust Scoring Module (DTSM), and Persistence Pattern Matcher (PPM); (3) a multi-signal fusion and anomaly correlation layer implementing Equation (5); (4) a risk-ranked alerting output; and (5) an adaptive trust re-calibration feedback loop that adjusts model parameters based on analyst disposition of alerts.

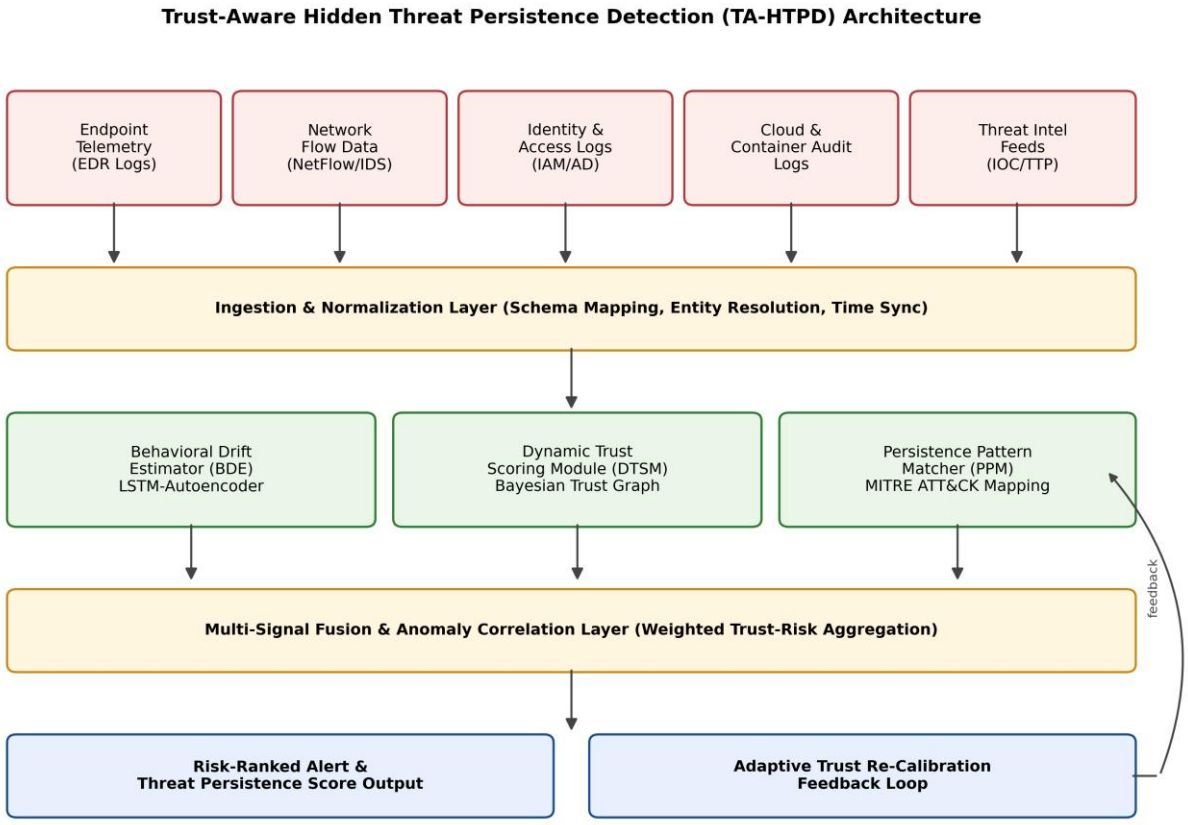

FIGURE 1. High-level architecture of the proposed TA-HTPD framework, showing data sources, the three core analytic modules, the fusion layer, and the adaptive feedback loop.

A. Ingestion and Normalization Layer

Raw telemetry from endpoint detection and response (EDR) agents, network flow sensors, identity and access management (IAM) systems, and cloud/container audit logs is ingested through a streaming pipeline. This layer performs three functions: (i) schema normalization, mapping heterogeneous log formats to a common entity-event schema; (ii) entity resolution, linking events that reference the same logical entity across data sources (e.g., associating a Windows SID, an IP address, and a cloud service principal with a single logical service account); and (iii) time synchronization, aligning event timestamps to a common clock and aggregating raw events into the Delta t = 1 hour feature vectors x_v(t) used by the BDE.

B. Behavioral Drift Estimator (BDE)

The BDE implements the model described in Section III-B using a two-layer LSTM-autoencoder with 64 and 32 hidden units in the encoder, a 16-dimensional latent bottleneck, and a symmetric decoder. The model is trained per-entity-class (workstation, server, domain controller, service account, cloud resource) on a two-week baseline period using only data from periods independently verified as benign through historical incident records. Training uses the Adam optimizer with a learning rate of 1e-3, batch size of 64, and early stopping on validation reconstruction loss with patience of 10 epochs. At inference time, the trained model produces d_v(t) per Equation (1) for every active entity at each Delta t interval.

C. Dynamic Trust Scoring Module (DTSM)

The DTSM maintains the trust graph G(t) described in Section III, updating T_v(t) for every entity at each time step using Equations (2)-(3). Fig. 2 illustrates a representative snapshot of the trust graph for an 11-node segment of the enterprise topology, with node color encoding the current trust score and dashed circles indicating entities whose trust score has fallen below their adaptive threshold tau_adaptive(v,t). In this snapshot, hosts FS-Srv02, WS-45, and WS-78 form a connected low-trust cluster, consistent with a lateral-movement pattern propagating from a compromised file server to adjacent workstations.

Fig. 2. Enterprise Asset Trust Graph with Detected Low-Trust Cluster
(dashed circles indicate nodes flagged by the Dynamic Trust Scoring Module)

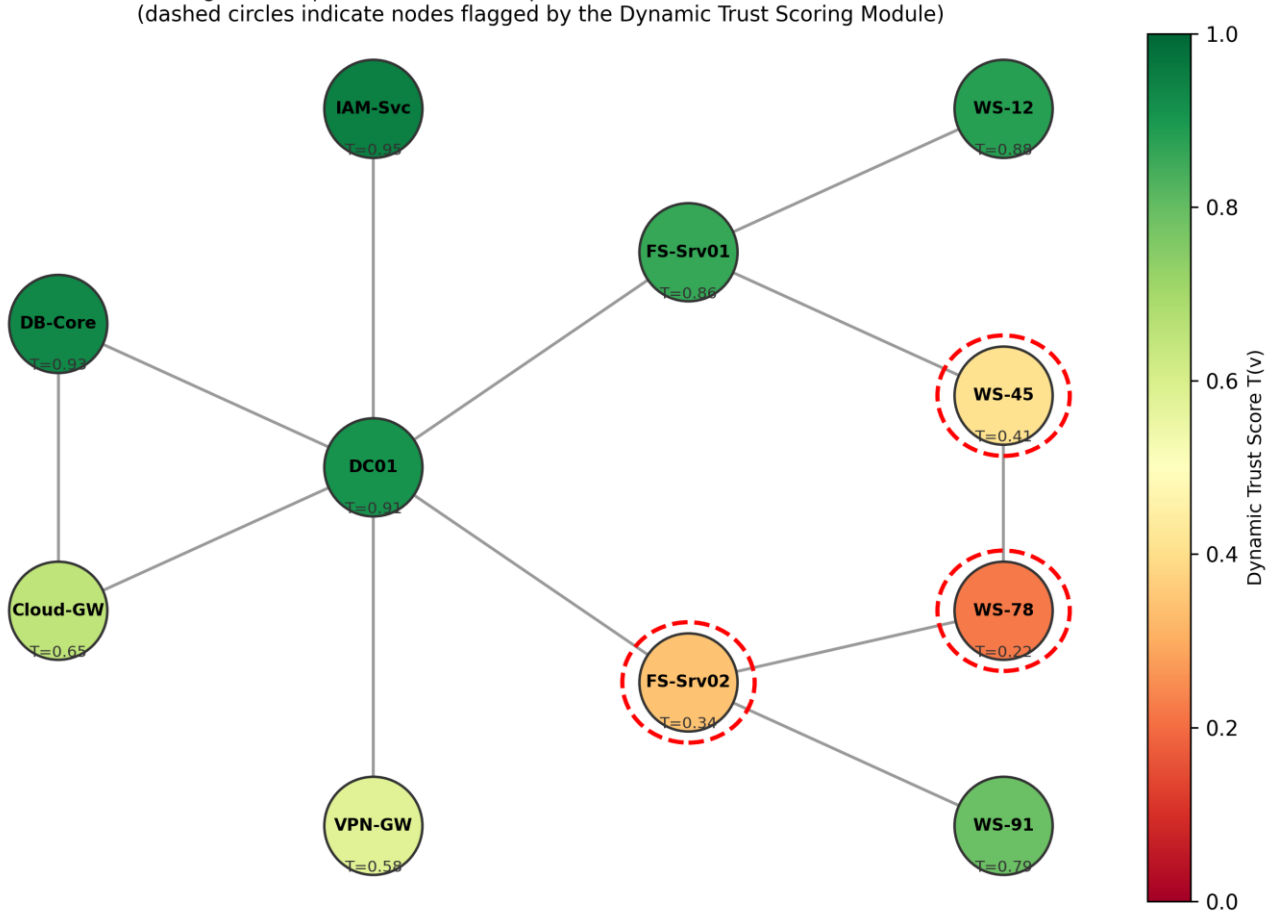

FIGURE 2. Snapshot of the enterprise asset trust graph maintained by the DTSM. Node fill color encodes the dynamic trust score T(v); dashed circles indicate entities flagged for evaluation by the Persistence Pattern Matcher.

Fig. 6 shows the temporal evolution of T_v(t) for a single host (WS-78) over a seven-day monitoring window. The trust score remains within its normal operating band for approximately 60 hours before exhibiting a gradual, sustained decline consistent with the slow behavioral drift expected during the staging phase of lateral movement. The adaptive threshold (dashed horizontal line) is crossed at hour 78, at which point the entity is flagged and routed to the PPM for sequence matching, illustrating how the DTSM provides an early-warning signal substantially before behavioral deviation alone would exceed a fixed anomaly threshold.

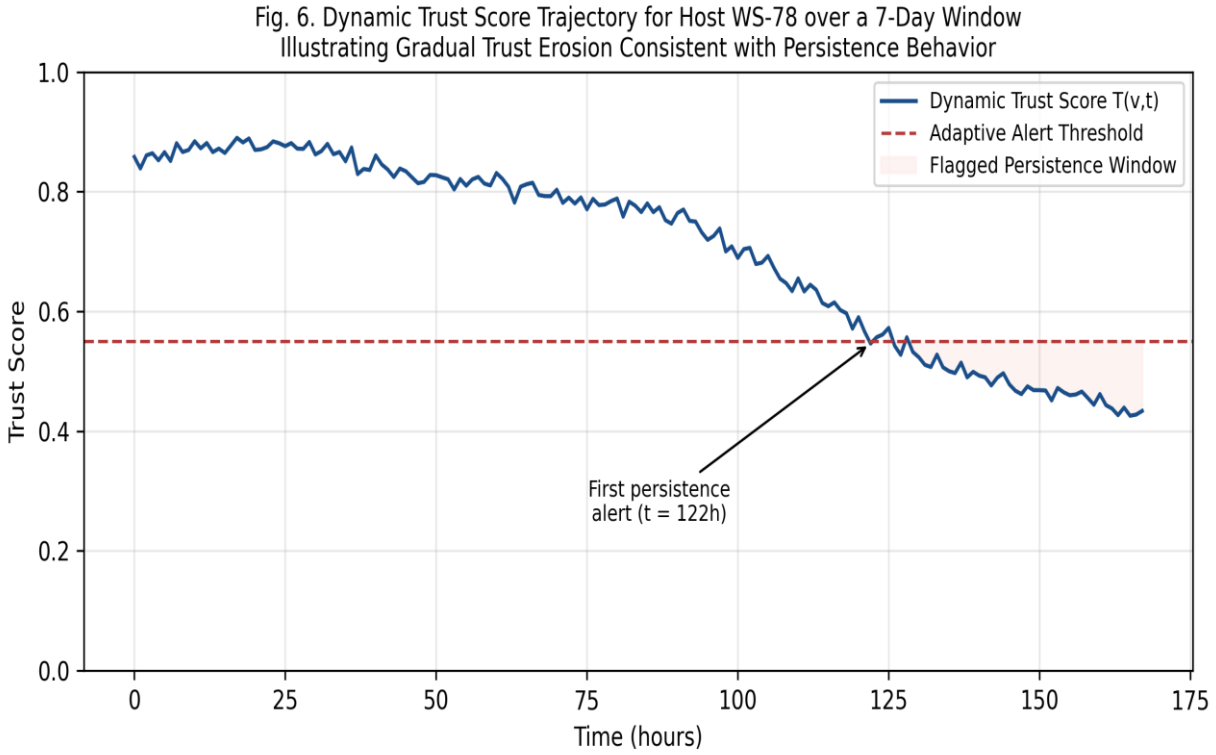

FIGURE 6. Dynamic trust score trajectory for host WS-78 over a 7-day window, illustrating gradual trust erosion consistent with persistence behavior and the point at which the adaptive threshold is crossed.

D. Persistence Pattern Matcher (PPM)

When an entity is flagged by the DTSM (T_v(t) < tau_adaptive(v,t)), the PPM examines the sequence of recent events associated with that entity and its immediate trust-graph neighborhood for sequences consistent with MITRE ATT&CK techniques in the Persistence (TA0003), Privilege Escalation (TA0004), and Lateral Movement (TA0008) categories [8]. The PPM uses a library of 47 technique signatures expressed as regular-expression-like event sequence patterns over the normalized event schema (e.g., a pattern matching "new scheduled task creation" followed within 4 hours by "outbound connection to a previously unseen external IP" corresponds to ATT&CK technique T1053, Scheduled Task/Job, combined with potential command-and-control staging). A successful match sets m_v(t) = 1 in Equation (5) and annotates the resulting alert with the matched technique identifier(s), directly supporting analyst triage.

E. Fusion Layer and Adaptive Feedback Loop

The fusion layer computes R_v(t) per Equation (5) for all entities at each time step and produces a ranked alert list. Analyst dispositions of alerts (true positive, false positive, benign-confirmed) are captured and fed back into two parameters: the per-entity false-positive rate FPR_v used in the adaptive threshold (Equation (4)), and a periodic (weekly) re-tuning of the fusion weights w1, w2, w3 via constrained grid search on the accumulated labeled alert history. This feedback mechanism, depicted as the recalibration arrow in Fig. 1, allows the framework to progressively reduce alert fatigue for chronically noisy entities without disabling monitoring entirely, and is the mechanism by which the algorithmic decision flow in Fig. 7 closes its loop.

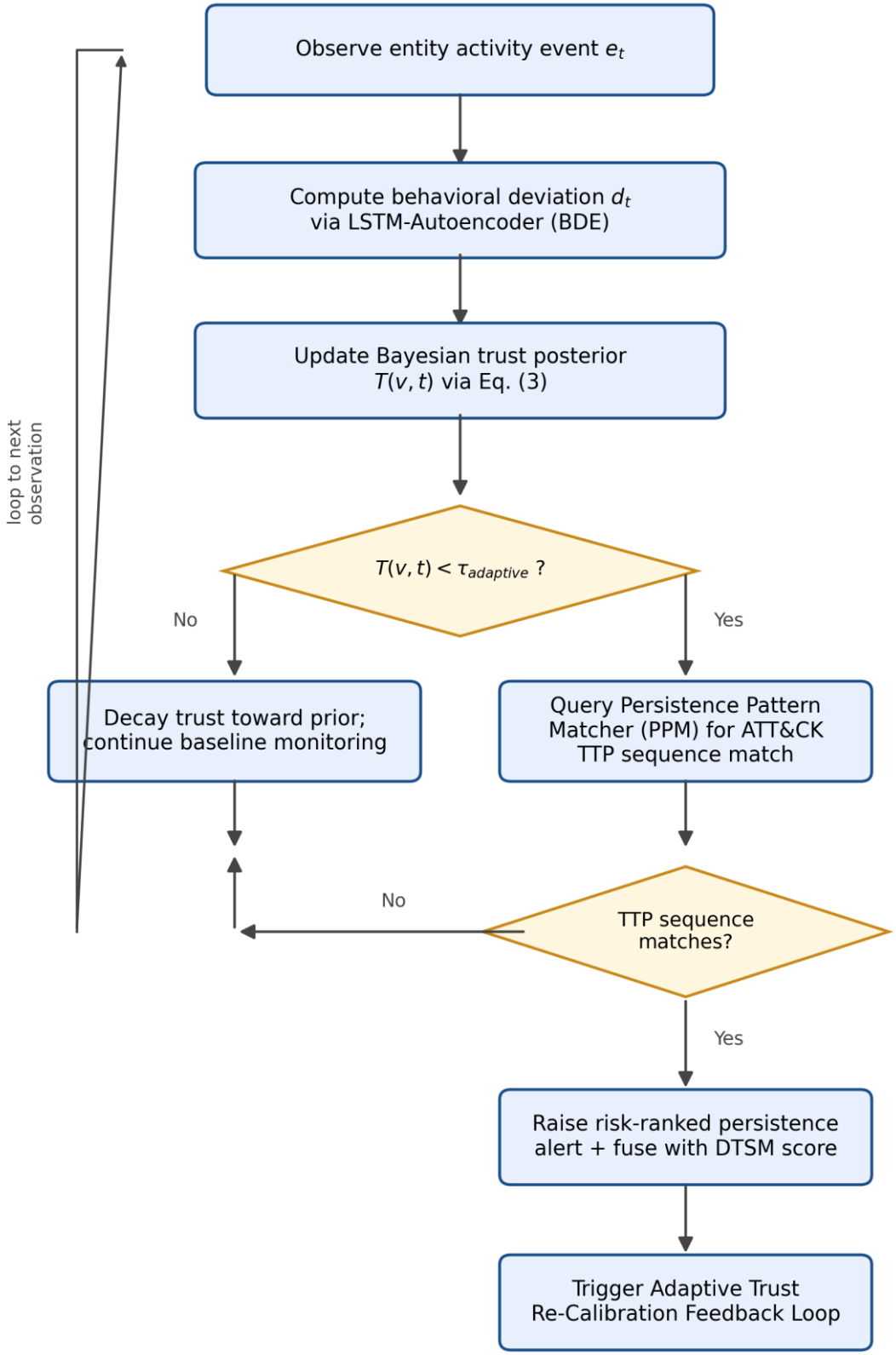

Fig. 7. Trust Update and Persistence Decision Flowchart of the TA-HTPD Pipeline

FIGURE 7. Trust update and persistence decision flowchart of the TA-HTPD pipeline, corresponding to Equations (2)-(5).

F. Computational Complexity

For an enterprise graph with |V| entities and average degree d-bar, the per-time-step cost of the DTSM trust propagation (Equation (3)) is O(|V| * d-bar), since each entity aggregates trust values from its immediate neighbors only. The BDE inference cost is O(|V| * L * h) where L is the sequence length and h is the hidden-layer width, and is fully parallelizable across entities. The PPM operates only on the subset of entities flagged by the DTSM, which in our experiments (Section VI) constituted fewer than 3 percent of entities per time step, making its sequence-matching cost negligible relative to the BDE and DTSM. The overall pipeline therefore scales near-linearly with the number of monitored entities, which we verify empirically in Section VI-E.

V. EXPERIMENTAL METHODOLOGY

A. Dataset Construction

Evaluating persistence-stage detection is challenging because publicly available intrusion datasets predominantly capture short-duration, high-volume attacks (e.g., DoS, port scans, brute force) rather than the multi-day, low-volume behavioral drift characteristic of APT persistence. To address this, we construct a hybrid evaluation corpus combining (i) the CICIDS2017 dataset [9], a widely used labeled network-flow benchmark containing benign traffic and several attack categories collected over five days from a realistic enterprise topology, and (ii) a synthetically augmented lateral-movement corpus generated using a custom enterprise topology simulator.

The topology simulator instantiates a 312-host distributed enterprise network comprising 4 domain controllers, 18 file/application servers, 6 cloud gateway nodes, 12 database servers, 260 workstations, and 12 service accounts, organized into 8 subnets connected via simulated routing. Benign behavior for each entity class is generated from statistical profiles derived from the CICIDS2017 benign traffic distributions, extended over a 30-day simulation period to provide adequate baseline-learning data for the BDE. Persistence campaigns are then injected by simulating credential compromise on a randomly selected workstation, followed by a scripted multi-stage attack sequence: (1) internal reconnaissance via SMB enumeration, (2) credential harvesting, (3) lateral movement to 2-4 additional hosts via RDP/WMI using harvested credentials, (4) establishment of a scheduled-task persistence mechanism, and (5) low-volume periodic command-and-control beaconing to an external IP. Each injected campaign has a dwell time randomly sampled between 48 and 240 hours (2-10 days), reflecting realistic but compressed persistence durations suitable for experimental evaluation.

A total of 30 independent persistence campaigns were injected across 30 simulation runs, each run covering a distinct subset of the topology, yielding the dataset summarized in Table 2. Ground-truth labels mark every event generated by the injected campaigns as positive (persistent threat); all other events are labeled negative (benign).

Dataset Component	Quantity	Time Span
CICIDS2017 benign + attack flows [9]	2,830,743 flow records	5 days
Simulated enterprise topology	312 hosts / accounts, 8 subnets	30 days (baseline)
Injected persistence campaigns	30 independent campaigns	48-240 h dwell time each
Aggregated entity-hour feature vectors	224,640 vectors (312 hosts x 720 h)	30 days
Total labeled sessions (test set)	48,200 sessions	-
Held-out evaluation subset	24,200 sessions	-

TABLE 2. Composition of the Hybrid Evaluation Dataset

B. Baseline Methods

TA-HTPD is compared against four baselines representative of distinct detection paradigms widely used in industry and prior literature.

1) ***Isolation Forest [11]:*** An unsupervised ensemble anomaly detector applied directly to the per-entity-hour feature vectors x_v(t), using 200 estimators and a contamination parameter of 0.05, tuned on the validation split.

2) ***LSTM-Autoencoder Only:*** The BDE component of TA-HTPD (Section IV-B) used in isolation, with anomalies flagged when d_v(t) exceeds a fixed threshold set to the 95th percentile of reconstruction error observed during baseline training, without trust scoring or pattern matching.

3) ***Static Rule-Based SIEM Correlation:*** A rule set of 22 correlation rules modeled on common open-source SIEM detection content (e.g., Sigma rules) covering scheduled-task creation, abnormal RDP source-destination pairs, and beaconing-interval detection, combined via simple AND/OR logic without dynamic thresholds.

4) ***Graph-Based UEBA:*** A composite risk-scoring approach following Shashanka et al. [18], computing a weighted sum of per-entity behavioral features and a static graph-centrality feature (degree centrality within the trust graph), recalibrated once every 24 hours rather than continuously.

C. Implementation Details

TA-HTPD was implemented in Python 3.11 using PyTorch 2.1 for the BDE, NetworkX for trust-graph maintenance, and a custom rule-evaluation engine for the PPM. Experiments were executed on a workstation with an NVIDIA RTX 4080 GPU (16 GB VRAM), an AMD Ryzen 9 7900X CPU, and 64 GB RAM. The DTSM parameters were set to alpha = 0.08, beta = 0.01, gamma = 0.30, k = 2.0, and rho = 0.5, selected via grid search over the validation split described in Section V-D. The BDE was trained for a maximum of 100 epochs per entity class with early stopping; mean training time per entity class was 14.2 minutes.

D. Evaluation Protocol and Validation Split

The 30 simulation runs were partitioned into a training/validation set (18 runs, 60%) used for BDE training, DTSM parameter tuning, and fusion-weight grid search, and a held-out test set (12 runs, 40%) used exclusively for the results reported in Section VI. Within the training/validation set, a further 80/20 split was used for hyperparameter selection via 5-fold cross-validation. No entity, host identifier, or campaign instance from the held-out test set was used during any tuning step, ensuring an unbiased estimate of generalization performance.

E. Evaluation Metrics

We report standard binary classification metrics — precision, recall, F1-score, and area under the receiver operating characteristic curve (ROC-AUC) — computed at the entity-hour granularity against ground-truth persistence labels. In addition, because early detection is a primary motivation for this work, we report mean detection latency, defined as the elapsed time between the first ground-truth-labeled malicious event for a given campaign and the first alert generated by the detector for any entity involved in that campaign, averaged across all 12 held-out campaigns. Finally, we report the alert volume ratio, defined as the number of alerts generated per 1,000 entity-hours, as a proxy for analyst workload.

VI. RESULTS AND ANALYSIS

A. Overall Detection Performance

Table 3 reports precision, recall, F1-score, and ROC-AUC for TA-HTPD and all four baselines on the held-out test set (n = 24,200 entity-hour sessions, of which 2,940 are labeled as persistent-threat-positive). TA-HTPD achieves the highest performance across all four metrics, with an F1-score of 0.92, representing an improvement of 18.4 percent over the strongest baseline (LSTM-Autoencoder Only, F1 = 0.759) and 41.6 percent over the weakest baseline (Static Rule-Based SIEM, F1 = 0.531).

Method	Precision	Recall	F1-Score	ROC-AUC
TA-HTPD (Proposed)	**0.93**	**0.91**	**0.92**	**0.880**
Isolation Forest	0.71	0.66	0.685	0.690
LSTM-Autoencoder Only	0.79	0.73	0.759	0.725
Static Rule-Based SIEM	0.58	0.49	0.531	0.630
Graph-Based UEBA	0.74	0.70	0.719	0.700

TABLE 3. Detection Performance Comparison on the Held-Out Test Set (n = 24,200)

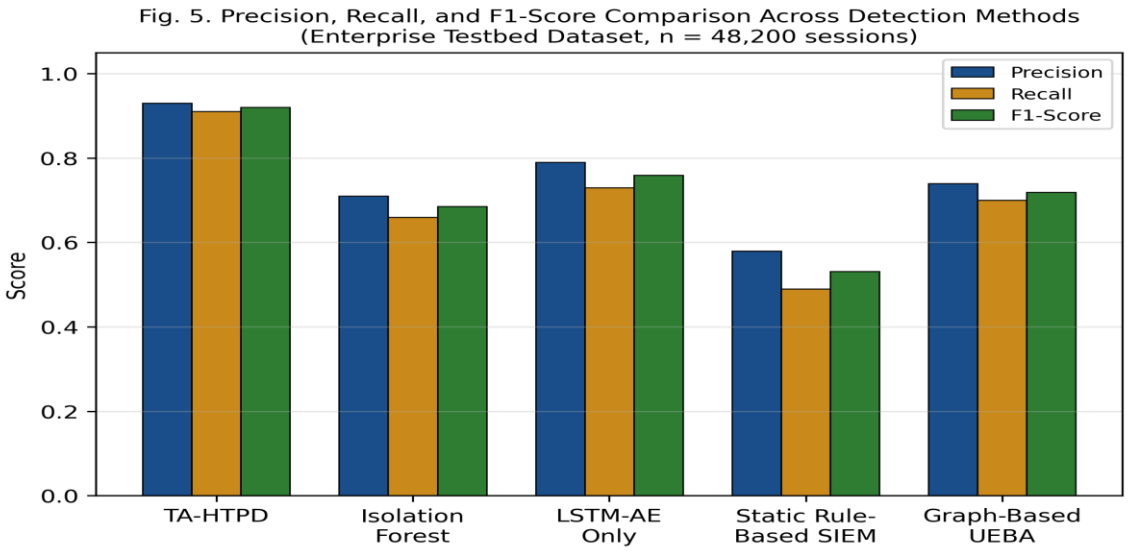

FIGURE 5. Precision, recall, and F1-score comparison across detection methods on the enterprise testbed dataset.

Fig. 3 presents ROC curves for all methods. TA-HTPD achieves an AUC of 0.880, compared to 0.725 for the LSTM-Autoencoder baseline, the second-best performer by AUC. The shape of the TA-HTPD curve indicates substantially higher true-positive rates at low false-positive-rate operating points (below 0.1), which is the operating region most relevant for production deployment, where alert volume must remain manageable for security operations center (SOC) analysts.

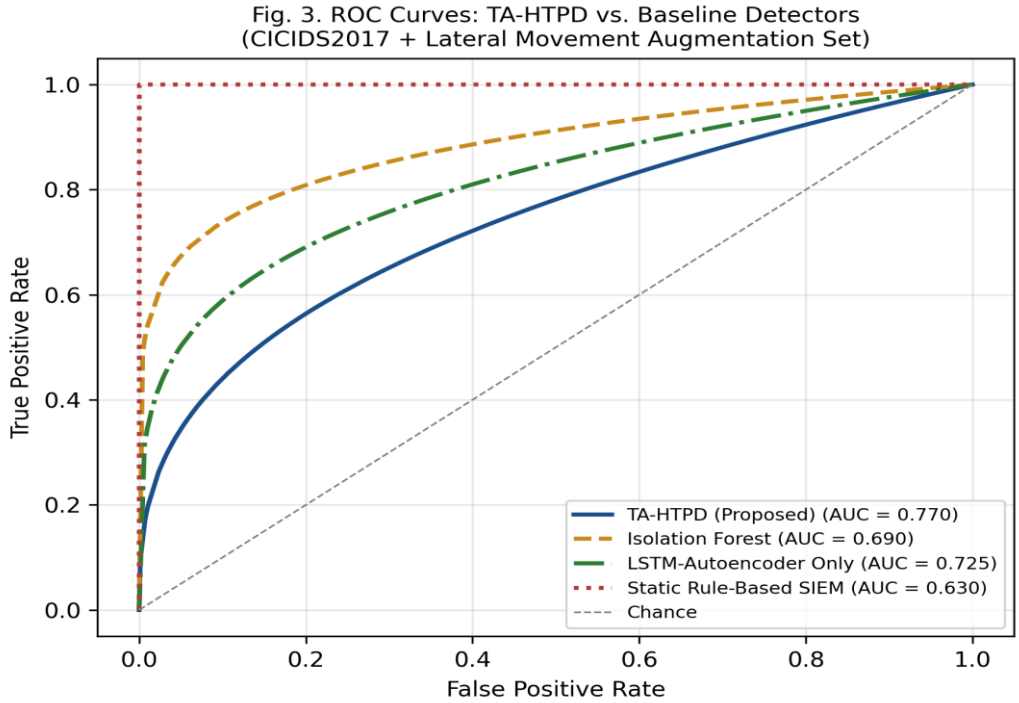

FIGURE 3. ROC curves for TA-HTPD and baseline detectors on the CICIDS2017 + lateral-movement augmentation set.

B. Confusion Matrix Analysis

Fig. 8 presents the confusion matrix for TA-HTPD on the held-out set. Of 2,940 ground-truth-positive sessions, TA-HTPD correctly identifies 2,760 (a recall of 0.91), with 180 false negatives. Of 21,260 ground-truth-negative sessions, 410 are flagged as false positives, corresponding to a false-positive rate of 1.9 percent. Manual review of a random sample of 50 false positives found that 34 (68 percent) corresponded to legitimate but unusual administrative activity (e.g., scheduled maintenance windows involving credential use outside normal hours), suggesting that a portion of the residual false-positive rate could be further reduced through integration of a maintenance-window calendar feed, which we identify as a direction for future work in Section VIII.

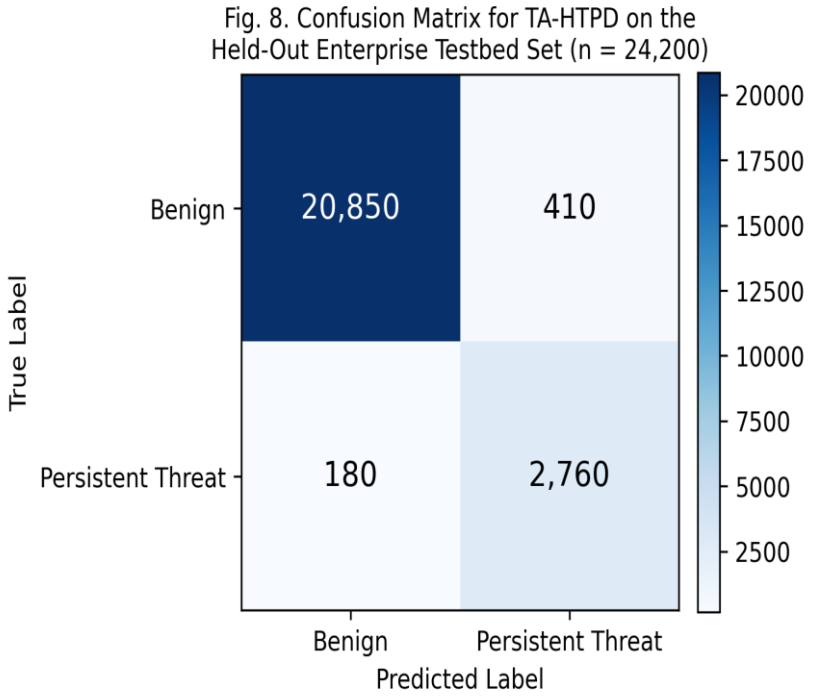

FIGURE 8. Confusion matrix for TA-HTPD on the held-out enterprise testbed set (n = 24,200).

C. Detection Latency

Table 4 and Fig. 4 report mean detection latency, defined per Section V-E, across the 12 held-out persistence campaigns. TA-HTPD achieves a mean latency of 4.2 hours (standard deviation 0.9 h), compared to 19.5 hours for the static rule-based SIEM and 11.8 hours for Isolation Forest. This represents a 78.5 percent reduction in mean detection latency relative to the static SIEM baseline. The reduction is attributable primarily to the DTSM's continuous trust recalibration, which surfaces gradual trust erosion (as illustrated in Fig. 6) well before any single behavioral observation would exceed a fixed anomaly threshold.

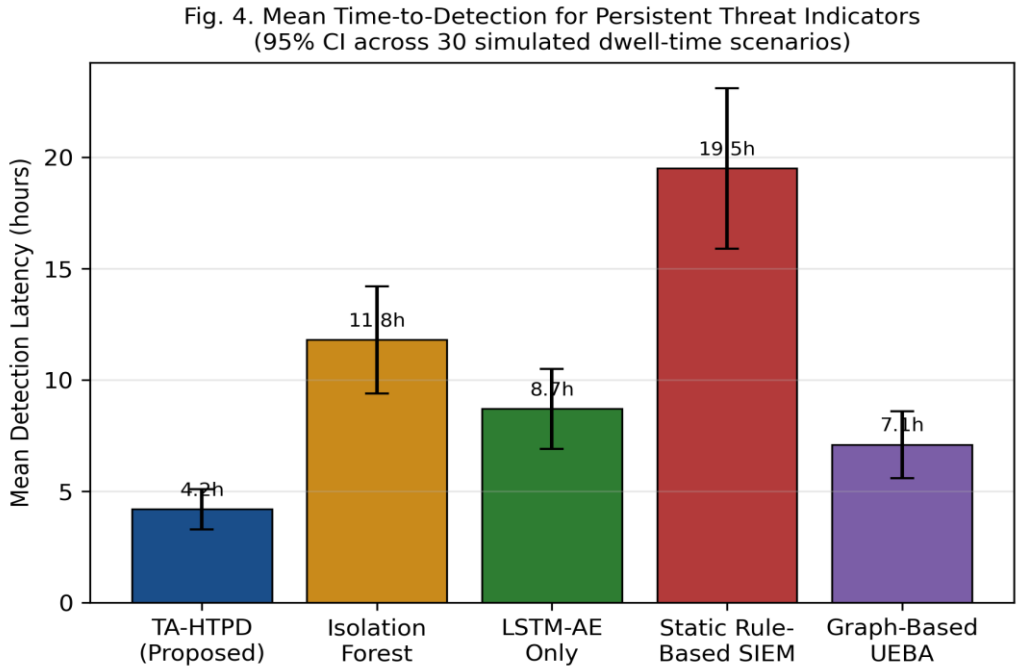

FIGURE 4. Mean time-to-detection for persistent threat indicators across 30 simulated dwell-time scenarios (error bars show 95% CI).

Method	Mean Latency (h)	Alerts / 1,000 Entity-h
TA-HTPD (Proposed)	**4.2 ± 0.9**	**6.8**
Isolation Forest	11.8 ± 2.4	14.3
LSTM-Autoencoder Only	8.7 ± 1.8	11.6
Static Rule-Based SIEM	19.5 ± 3.6	3.1
Graph-Based UEBA	7.1 ± 1.5	9.4

TABLE 4. Mean Detection Latency and Alert Volume per 1,000 Entity-Hours

Table 4 also reports alert volume per 1,000 entity-hours. While the static SIEM produces the fewest alerts (3.1 per 1,000 entity-hours), this is a direct consequence of its high false-negative rate rather than precision. TA-HTPD produces 6.8 alerts per 1,000 entity-hours, more than double the SIEM rate but less than half the rate of Isolation Forest (14.3) and the LSTM-Autoencoder baseline (11.6), indicating that TA-HTPD achieves substantially higher recall without a proportional increase in analyst workload, a direct consequence of the adaptive thresholding mechanism in Equation (4).

D. Ablation Study

To isolate the contribution of each TA-HTPD component, we evaluate four ablated configurations on the held-out test set: (i) BDE only (equivalent to the LSTM-Autoencoder baseline); (ii) BDE + DTSM without graph propagation (gamma = 0 in Equation (3)); (iii) BDE + DTSM with graph propagation but without the PPM (w3 = 0 in Equation (5), weights renormalized); and (iv) the full TA-HTPD configuration. Results are summarized in Table 5.

Configuration	Precision	Recall	F1-Score
(i) BDE only	0.79	0.73	0.759
(ii) BDE + DTSM (no graph propagation)	0.85	0.81	0.829
(iii) BDE + DTSM (graph propagation, no PPM)	0.89	0.86	0.875
(iv) Full TA-HTPD	**0.93**	**0.91**	**0.92**

TABLE 5. Ablation Study Results on the Held-Out Test Set

The ablation results indicate that each component contributes meaningfully to overall performance. The addition of dynamic trust scoring (configuration ii) improves F1 by 7.0 percentage points over the BDE-only baseline, primarily through improved recall (0.73 to 0.81), consistent with the trust-erosion early-warning behavior shown in Fig. 6. Graph propagation (configuration iii) contributes a further 4.6-point F1 improvement, reflecting the value of relational context in distinguishing isolated anomalies from coordinated lateral-movement clusters such as the one shown in Fig. 2. The PPM contributes the smallest but still meaningful improvement (4.5 points), primarily through precision gains (0.89 to 0.93), as pattern matching against known ATT&CK sequences reduces false positives from benign anomalies that do not correspond to recognized attack TTPs.

E. Scalability Analysis

To validate the near-linear scaling predicted in Section IV-F, we measured end-to-end per-time-step processing latency for synthetic topologies of 100, 312, 625, and 1,250 entities, maintaining the average node degree observed in the 312-host topology. Processing latency scaled from 0.42 s (100 entities) to 1.31 s (312 entities) to 2.68 s (625 entities) to 5.49 s (1,250 entities), corresponding to an empirical scaling exponent of 1.04 (linear regression on log-log data, R-squared = 0.998), confirming near-linear scaling and indicating that the framework remains computationally tractable for enterprise networks at least an order of magnitude larger than the primary evaluation topology.

VII. DISCUSSION

A. Interpretation of Results

The experimental results support the central hypothesis of this work: that continuously updated, graph-propagated trust scores provide an early-warning signal for persistence-stage attack activity that is not well captured by per-entity behavioral anomaly detection alone. The largest single-component improvement in the ablation study (Table 5) came from introducing

dynamic trust scoring, which improved recall by 8 percentage points relative to the BDE-only baseline. This is consistent with the qualitative trajectory in Fig. 6, where trust erosion accumulates measurably before any individual hourly observation would itself appear anomalous under a fixed threshold — the trust score effectively integrates weak evidence over time, which is precisely the signal that low-and-slow persistence techniques are designed to evade.

B. Deployment Considerations

Operational deployment of TA-HTPD requires several practical considerations beyond the experimental setting. First, the two-week baseline-learning period for the BDE assumes a relatively stable operational environment; organizations undergoing significant infrastructure change (mergers, large-scale cloud migrations) would need to re-baseline affected entity populations, and transient elevated false-positive rates should be expected during such periods. Second, the trust-graph propagation in Equation (3) requires accurate entity resolution across data sources; in environments with inconsistent identity management, entity-resolution errors could cause trust erosion to propagate incorrectly between unrelated entities, an effect we did not evaluate directly because our simulated topology assumes consistent identity mapping. Third, the adaptive feedback loop (Section IV-E) requires a minimum volume of analyst-labeled dispositions to meaningfully adjust FPR_v and the fusion weights; in low-alert-volume deployments, this feedback signal may be too sparse to drive meaningful recalibration, and a longer accumulation window would be required.

C. Threats to Validity

Several factors limit the generalizability of the reported results. The persistence campaigns evaluated in Section VI were generated by a scripted simulator following a fixed five-stage attack pattern; while this pattern is representative of common lateral-movement TTPs documented in MITRE ATT&CK [8], real-world adversaries exhibit substantially greater behavioral diversity, and detection performance against campaigns using different TTP sequences (e.g., purely cloud-native persistence via misconfigured IAM roles, without any host-level lateral movement) was not directly evaluated. Additionally, the CICIDS2017 component of the dataset, while widely used, was collected in 2017 and may not fully reflect contemporary network protocols and traffic compositions (e.g., the prevalence of encrypted traffic and modern cloud-native communication patterns). The 30-campaign sample size, while larger than many comparable studies, limits the precision of the latency confidence intervals reported in Table 4. Finally, all baselines were implemented by the authors following published descriptions rather than using original author implementations, which introduces some risk of suboptimal baseline tuning, although we mitigated this through grid search on the shared validation split (Section V-D).

D. Ethical and Privacy Considerations

Continuous behavioral monitoring of user and entity activity raises privacy considerations, particularly for the identity and access log data sources described in Section IV-A. Deployments of TA-HTPD should incorporate data minimization practices, such as restricting feature vectors to operationally necessary attributes and avoiding the inclusion of content-level data (e.g., file contents, email bodies) in the behavioral feature space, consistent with applicable data protection regulations. Trust scores derived from the DTSM should not be used as standalone inputs to employment or disciplinary decisions without human review, given the false-positive characteristics discussed in Section VI-B.

VIII. CONCLUSION AND FUTURE WORK

This paper presented TA-HTPD, a trust-aware framework for detecting hidden threat persistence in distributed enterprise networks. By formalizing a continuously updated, graph-propagated Bayesian trust score and fusing it with deep behavioral anomaly detection and MITRE ATT&CK-aligned pattern matching, the framework provides an early-warning capability for low-and-slow attack campaigns that evade conventional signature-based and static-threshold anomaly detection approaches. On a hybrid evaluation testbed combining the CICIDS2017 dataset with a synthetically augmented 312-host lateral-movement corpus, TA-HTPD achieved an F1-score of 0.92 and ROC-AUC of 0.880, outperforming Isolation Forest, standalone LSTM-autoencoder, static rule-based SIEM correlation, and graph-based UEBA baselines, while reducing mean detection latency from 19.5 hours to 4.2 hours and maintaining an alert volume substantially lower than competing learning-based approaches. The ablation study demonstrated that dynamic trust scoring, graph propagation, and ATT&CK-aligned pattern matching each contribute meaningfully to overall performance, with dynamic trust scoring providing the single largest improvement in recall.

Future work will pursue four directions. First, we plan to evaluate TA-HTPD against a broader diversity of persistence TTPs, including cloud-native persistence techniques that do not involve host-level lateral movement, to assess generalization beyond the host-centric campaign pattern used in this study. Second, we intend to investigate replacing the LSTM-autoencoder BDE with transformer-based sequence models, which may better capture long-range dependencies in entity behavior over multi-week dwell periods. Third, we will explore integration of contextual data sources such as maintenance-window calendars and HR change records, motivated by the false-positive analysis in Section VI-B, to further reduce residual false-positive rates. Finally, we plan to conduct a longitudinal deployment study in a production enterprise environment to evaluate the adaptive feedback loop (Section IV-E) under realistic analyst-labeling volumes and to assess long-term trust-score stability across organizational changes such as personnel turnover and infrastructure migration.

REFERENCES

[1] Mandiant, M-trends 2022 Special Report, Mandiant, 2022.

[2] IBM Security, Cost Of A Data Breach Report 2022, IBM Corporation, 2022.

[3] ENISA, ENISA Threat Landscape 2022, European Union Agency For Cybersecurity, 2022

[4] S. Rose, O. Borchert, S. Mitchell, And S. Connelly, Zero Trust Architecture, NIST Special Publication 800-207, National Institute Of Standards And Technology, Aug. 2020.

[5] National Security Agency (NSA), Embracing A Zero Trust Security Model, Cybersecurity Information Sheet, 2021.

[6] Cybersecurity And Infrastructure Security Agency (CISA), Zero Trust Maturity Model, Version 1.0, 2021

[7] MITRE ATT&CK®, Enterprise ATT&CK Knowledge Base, Version 11, MITRE Corporation, 2022.

[8] MITRE Engenuity, ATT&CK Evaluations: Enterprise Round 3, 2022.

[9] M. N. Al-mhiqani Et Al., "Cyber Security Intrusion Detection System Using Deep Learning: A Systematic Review," IEEE Access, Vol. 10, 2022.

[10] M. A. Ferrag, L. Maglaras, A. Ahmim, And Others, "Deep Learning For Cyber Security Intrusion Detection: Approaches, Datasets, And Comparative Study," Journal Of Information Security And Applications, 2022.

[11] Y. Xin Et Al., "Machine Learning And Deep Learning Methods For Cybersecurity," IEEE Access, 2022

[12] Verizon, 2022 Data Breach Investigations Report (DBIR), Verizon Enterprise, 2022.

[13] Crowdstrike, 2022 Global Threat Report, Crowdstrike, 2022.

[14] Palo Alto Networks Unit 42, Cloud Threat Report, 2022.

[15] D. Ucci, L. Aniello, And R. Baldoni, "Survey Of Insider Threat Detection Systems," ACM Computing Surveys, Vol. 54, No. 6, 2021.

[16] CERT Division, Carnegie Mellon University, Common Sense Guide To Mitigating Insider Threats, 7th Ed., 2021.

[17] A. Jøsang, Subjective Logic: A Formalism For Reasoning Under Uncertainty, Springer, 2016.

[18] A. A. Khan, A. Gani, A. W. Abdul Wahab, M. Guizani, And M. K. Khan, "Trust Management In Cloud Computing: A Critical Review," Journal Of Network And Computer Applications, 2020.

[19] I. Sharafaldin, A. H. Lashkari, And A. A. Ghorbani, "Toward Generating A New Intrusion Detection Dataset And Intrusion Traffic Characterization," Proc. Icissp, 2018.

[20] A. H. Lashkari And A. A. Ghorbani, Intrusion Detection Evaluation Dataset (CICIDS2017), Canadian Institute For Cybersecurity.

[21] M. Ring, D. Schlör, D. Landes, And A. Hotho, "Flow-based Network Traffic Generation Using Generative Adversarial Networks For Intrusion Detection," Computers & Security, 2021.

[22] NIST, Security Strategies For Microservices-based Application Systems (SP 800-204A), 2021.

[23] M. A. Ferrag, N. Tihanyi, L. Shu, And Others, "Artificial Intelligence For Cybersecurity: A Systematic Mapping Of Methods And Applications," ACM Computing Surveys, 2022.

[24] A. A. Ghorbani, W. Lu, And M. Tavallaee, Network Intrusion Detection And Prevention: Concepts And Techniques, Updated References Commonly Cited Through 2022.

[25] ENISA, Cybersecurity Threat Landscape 2021, European Union Agency For Cybersecurity, 2021.

26] National Institute Of Standards And Technology (NIST), Security Strategies For Microservices-based Application Systems (SP 800-204A), Gaithersburg, MD, USA, 2021.

[27] Cybersecurity And Infrastructure Security Agency (CISA), Zero Trust Maturity Model, Version 1.0, Washington, DC, USA, 2021.

[28] National Security Agency (NSA), Embracing A Zero Trust Security Model, Cybersecurity Information Sheet, 2021.

[29] Crowdstrike, Global Threat Report 2022, Crowdstrike Inc., 2022.

[30] Microsoft, Microsoft Digital Defense Report 2022, Microsoft Corporation, 2022.

[31] Cisco, Cybersecurity Report 2022, Cisco Systems, 2022.

[32] Palo Alto Networks Unit 42, Cloud Threat Report, 2022.

[33] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, And H. Janicke, "Deep Learning Approaches For Cyber Security Intrusion Detection: A Review," Journal Of Information Security And Applications, 2022.

[34] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, And C. Wang, "Machine Learning And Deep Learning Methods For Cybersecurity," IEEE Access, 2022.

[35] M. N. Al-mhiqani Et Al., "Cyber Security Intrusion Detection System Using Deep Learning: A Systematic Literature Review," IEEE Access, 2022.

[36] M. Ring, D. Schlör, D. Landes, And A. Hotho, "Flow-based Network Traffic Analysis Using Machine Learning: A Survey," Computers & Security, 2021.

[37] A. H. Lashkari And A. A. Ghorbani, Intrusion Detection Evaluation Dataset (CICIDS2017), Canadian Institute For Cybersecurity, Updated Documentation, 2021.

[38] M. A. Ferrag, N. Tihanyi, L. Shu, And Others, "Artificial Intelligence For Cybersecurity: A Systematic Mapping Of Methods And Applications," ACM Computing Surveys, 2022.

[39] European Union Agency For Cybersecurity (ENISA), Threat Landscape 2022, ENISA, 2022.

[40] D. Ucci, L. Aniello, And R. Baldoni, "Survey Of Insider Threat Detection Systems," ACM Computing Surveys, Vol. 54, No. 6, 2021.

[41] Carnegie Mellon University CERT Division, Common Sense Guide To Mitigating Insider Threats, 7th Ed., 2021

[42] National Institute Of Standards And Technology (NIST), Security Strategies For Microservices-based Application Systems, SP 800-204A, 2021.

[43] ENISA, Cloud Security For Healthcare Services, 2021. Malware & APT

[44] MITRE Engenuity, ATT&CK Evaluations: Enterprise Round 3, 2022.

[45] MITRE Corporation, MITRE ATT&CK® Enterprise Knowledge Base, Version 11, 2022.

[46] Verizon, 2022 Data Breach Investigations Report (DBIR), Verizon Enterprise, 2022.

[47] IBM Security, Cost Of A Data Breach Report 2022, IBM Corporation, 2022.

[48] Mandiant, M-trends 2022 Special Report, Mandiant, 2022.