SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Predictive Dynamic Attack-Path Intelligence for Proactive Cloud Threat Detection

Kalyani N [1]*, Sirisha Museboyina [2]

[1] Research scholar, Department of CSE, Malla Reddy University, Hyderabad, Telangana, India. ORCID: 0009-0004-0281-0135
[2] Associate Professor, Department of CSE, Malla Reddy University, Hyderabad, Telangana, India. ORCID: 0000-0002-9733-8870
* Corresponding author: Kalyani N, kalyaninabha1406@gmail.com

Abstract
Cloud-scale threat detection can reconstruct malicious activity after evidence becomes visible, but proactive defence requires reliable forecasts of where a partially observed intrusion is likely to progress. Predictive Dynamic Attack-Path Intelligence (PDAPI) is a proposed temporal graph framework in this paper, which localises changes of interest in security concerns, identifies partial attack-paths, and predicts future nodes, actions, targets of interest, and continuations of a multi-hop attack. PDAPI integrates temporal-structural encoding, feasibility-based beam search, calibrated on lead-time risk scoring, and evidence-based agentic validation. An empirical evaluation was constructed over temporally ordered attack-path sequences derived from provenance-oriented cloud-security traces and ATT&CK-aligned behaviours. Across the consolidated test split, PDAPI achieved Hit@1 of 0.873, Hit@3 of 0.957, mean reciprocal rank of 0.912, next-action macro-F1 of 0.901, and critical-target Top-3 accuracy of 0.934. It provided a median warning lead time of 18.7 min while maintaining 42 ms median prediction latency. Ablation results showed that removing temporal encoding, feasibility constraints, or changed-subgraph localisation reduced Hit@1 by 4.1, 5.8, and 3.4 percentage points, respectively. The findings confirm the use of predictive attack-path intelligence as a viable expansion to dynamic attack-graph detection to effectively enable proactive cloud defence.

Keywords - attack forecasting, cloud security, dynamic attack graph, proactive threat detection, temporal graph intelligence

1. Introduction

Cloud computing has transformed the performance of business protection. Service authentication is provided by identities, applications are decomposed into containers and serverless functions, infrastructure is programmatically created and destroyed, and a single user event can produce evidence in host, network, identity, storage, and control plane telemetry. These features increase the attack surface and complicate multi-stage intrusions to comprehend based on single alerts. Provenance-oriented security studies have thus modelled processes, files, sockets, users and other objects as graphs such that causal context is preserved along the duration of an activity. UNICORN demonstrated that long-term context can be retained by graph-based provenance to detect low-and-slow attacks, and THREATRACE shifted the analysis to graph learning at the node level to detect attacks early. [1,2] The more recent systems like MAGIC, KAIROS, TAPAS and ORTHRUS enhance self-supervised representation learning, reconstruction of the temporal features and online efficiency and quality of the attribution. [3–6]

The advancements answer a significant question: what is a malicious activity, and how does the evidence seen relate? They fail to completely answer another operational question, which will become decisive when the intrusion is merely visible: what is the next move that the attacker is likely going to make? Even when a suspicious script has

been run, a defender who determines the access to credentials has time to block further lateral movement, privileged use of cloud APIs, or an object store. The security value of only being able to recover the path which has already been travelled is not just that you can recover the path, but that you can be able to correctly estimate allowable future continuations sooner than they would have been otherwise in order to alter the course of events.

Probabilistic, sequence, knowledge-graph and temporal-graph methods have been used to predict attacks. Mohammadzad et al. modelled attacker behaviour dynamically, and with the help of a Hidden Markov Model, they could predict the subsequent host-level action; Dong et al. forecasted ATT&CK behaviours by using collaborative filtering and graph databases; Kuwano et al. used the unknown causal attack graph dynamics to update attack probabilities in cloud-based attack scenarios. [7–9] Current developments are

shifting to more sophisticated forecasting: physics-informed-graph neural networks for attack-path prediction, a hybrid heterogeneous graph paradigm that integrates spatial and temporal learning, transformer-based temporal graph network predict next attacker-victim interactions, and ATT&CK-based LSTM-Markov chains makes future predictions. [10–13] These experiments prove that reasoning to the future is possible, but indicate that there is a design defect between generic predictions of alerts and cloud-scale evidence-based path intelligence.

The main concept of this paper is that prediction ought to work with a changing partial attack path incorporated within a dynamic cloud graph. To be useful, a forecast must be causally possible, temporally possible, attributable based on available evidence, as well as operationally constrained. Based on this, the Predictive Dynamic Attack-Path Intelligence (PDAPI) takes each new observation as a graph update and localises the altered security region, derives an observed prefix and forecasts an ordered list of future extensions of the path. The prediction layer does not have the freedom to just make up actions: candidate transitions have to be justified by the reachability of the graph, by graph transition evidence that has been learned, by security restrictions or by external threat information. Uncertainty of the forecast grows with horizon, and is held in an explicit form, rather than bringing all predictions to just one deterministic trajectory.

The contribution is four times as much. The paper constructs proactive cloud threat detection to begin with as conditional future-path inference on a time-indexed heterogeneous attack graph. Second, it suggests a temporal-structural path encoder and a limited forecasting mechanism to predict the next-node, next-action, target, and multi-hop paths. Third, it presents lead-time-conscious predictive risk that simply takes into account probability, impact, feasibility, and uncertainty as well as the period to a critical state. Fourth, it fashions an evidence-limited agentic validation phase that transforms the predictions into regulated mitigation suggestions that do not permit spontaneous autonomous intervention. The resultant architecture is poised to be the predictive follow-up of reconstruction-oriented dynamic attack-graph pipelines. This transition, in a nutshell, is presented in Figure 1.

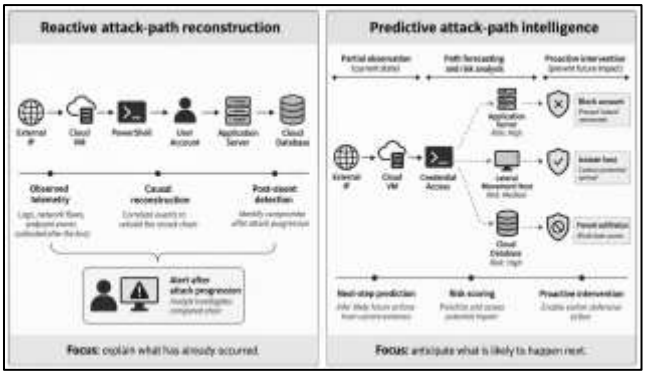

Fig. 1 Reactive attack-path reconstruction versus predictive attack-path intelligence.

2. Related Work

2.1. Provenance and Dynamic Attack-Graph Detection

The provenance graphs are the directed relationships among the security entities to describe the system execution. Causal continuity is the value of this representation: a strange process, ferreting out files, logging in, and network connections can be seen as part of a unified chain and not unrelated events. SLEUTH also showed reconstruction of real-time market settings based on audit information of commodities, and UNICORN subsequently provided provenance analysis of stealthy APTs on a long-horizon scale. [14,1] THREATRACE reconfigures GraphSAGE to learn benign entity roles and find threat-related nodes, and MAGIC reconfigures masked graph representation learning and model adaptation to lessen reliance on labelled attack data. [2,3] KAIROS is also informed of the temporal dynamics of provenance structure and subsequently restores compact attack footprints. [4]

Scalability is also now crucial. Incrementally maintained provenance graphs can have vast regions of invariant which do not require reprocessing when updating. TAPAS minimises spatial and temporal redundancy through task-guided graph segmentation and active-subgraph analysis, whereas ORTHRUS emphasises concise attack-path reconstruction and quality attribution. [5,6] Such systems encourage a change-conscious anticipatory design: the predictive part must not traverse the entire history graph to re-learn something, but will only need to reason based on all the recently changed regions of the graph and the causal neighbourhood it requires.

2.2. Temporal Graph Learning and Attack Forecasting

Future-link and future-event prediction have a natural computational framework based on temporal graph learning. TGAT learns time-based node representations with the help of time attention and functional time encoding, whereas Temporal Graph Networks take into consideration memory and message passing on event streams. [15,16] GraphSAGE and Graph Attention Networks are two alternative sources of local-structure learning mechanisms that involve an inductive and an attention-based approach, respectively. [17,18] These concepts have been applied to graphically changing instances of heterogeneous graphs and streams of attack alerts in security. Soylu and Das use breaststroke meta-path heterogeneous attention and recurrent temporal modelling to predict cyberattacks, and Nayeri and Rezvani build upon Temporal Graph Networks with edge classification, transformer aggregation, and dual-level temporal encoding to predict future interactions and attack types; DST-AttG is a combination of dynamic spatio-temporal attention and graph convolution with attack-chain inference and situational prediction. [11,12,20]

Attack-chain models (explicit attack-chains) are also covered by forecasting research. Mohammadzad et al. refresh a bare-bones attack graph on the fly and forecast the action that has just happened using an HMM. [7] Kuwano et al. use ATT&CK-based collaborative filtering to recommend likely subsequent behaviours. [9] Francois et al. discuss how full-path prediction is limited by dimensionality and the unsuitability of available datasets, which spurs graph-informed learning. [10] Raj et al. implement a hybrid of LSTM and Markov dynamics and beam search to produce risk-ranked chains of future chains based on ATT&CK. [13] Dong et al. apply uncertain causal graphs to back the dynamic reasoning of cloud risk. [8] All of these approaches prove to be predictive; however, a cloud deployment also needs to integrate with constantly evolving identities, workloads, resources, provenance, uncertainty, and the timing of interventions.

Table 1. Positioning of Representative Open-Access Approaches

Approach	Primary representation	Future prediction	Path / attribution	Online / dynamic
UNICORN [1]	Provenance graph	No	Contextual anomaly	Long-running
MAGIC [3]	Masked provenance graph	No	Multi-granularity	Adaptive
KAIROS [4]	Temporal provenance graph	No	Compact footprint	Yes
TAPAS [5]	Active task subgraph	No	Task-level	Yes
ORTHRUS [6]	Spatio-temporal provenance	No	Strong attribution	Yes
MAGD-HMM [7]	Dynamic minimal attack graph	Next action	Limited	Yes
DUCAG [8]	Uncertain causal graph	Risk / behaviour	Causal chain	Yes
ATT&CK CF [9]	Technique graph	Next behaviour	Technique-level	Offline/knowledge
TGNE-TA [12]	Temporal interaction graph	Victim + attack type	Interaction-level	Yes
PDAPI	Dynamic cloud attack graph	Next + multi-hop	Future path + intervention	Yes

3. Materials and Methods

3.1. Threat Model and Design Assumptions

PDAPI presumes the existence of a monitored cloud environment where security telemetry is provided by identity providers, virtual machines, containers, endpoint audit sources, network flows, cloud control planes, and resource-access logs. The attacker can gain a foothold either in the form of exposed services or stolen credentials, vulnerable workloads, malicious execution or control, or exfiltration and then conduct discovery, privilege escalation, credential access, and lateral movement, collection, command and control, or exfiltration. The model itself is not based on the assumption that all the malicious actions are explicitly labelled; instead, it requires that one step taken by the adversary results in at least one of the visible relations in the monitored telemetry. Hardware attacks are totally invisible side channels, and attacks that entirely compromise all telemetry sources prior to observation are out of scope.

The design is such that prediction and response authority are separated. High-impact actions can be subject to validation/policy controls regardless of forecasting recommending either containment or greater observation. It is important, as prediction is inherently uncertain. A proactive system which falsely quarantines the production services could inflict as much harm as the attack it is meant to prevent. PDAPI keeps forecast confidence, competing paths, and provenance of evidence as first-class output fields.

3.2. Dynamic Cloud Attack Graph

The monitored environment of time t is defined as a heterogeneous temporal graph G (t) which includes entities, directed relations, attributes and event time. Examples of nodes are users, workloads, processes, files, IP endpoints, cloud services, identity roles, secrets, and secured resources of the node. Examples of these operations include authenticates-to, executes, spawns, accesses, assumes-role, connects-to, invokes-api, reads, writes and downloads, which are examples of edges. New telemetry is received by updating the graph each time it is received.

$G_t = (V_t, E_t, X_t, T_t)$ (1)

Here V_t and E_t are the current node and edge sets, X_t stores entity and relation attributes, and T_t contains temporal information. For an incoming event e_(t+1), the update operator U modifies only affected entities and relations:

$G_{(t+1)} = U(G_t, e_{(t+1)})$ (2)

The transformed region ΔG t has new nodes, new edges, new attributes, and those whose security state has been altered. This incorporates a k- hop causal neighbourhood such that a new observation will have enough antecedent and successor context. This is in accordance with the principle of scalability of the online provenance systems, where to emphasize costly computation on active or security-relevant regions but not on the invariant graph. [5]

Table 2. Core PDAPI Graph Entities and Forecast-Relevant Attributes

Entity	Examples	Forecast-relevant attributes
Identity	user, service account, role	privilege, MFA state, trust relation, session age
Workload	VM, container, function	image, role, exposure, host, security group
Process	shell, script, binary	parent, executable, signer, command line
Resource	database, bucket, key vault	criticality, sensitivity, access policy
Network	IP, endpoint, flow	direction, port, protocol, reputation
Event	login, API call, access	timestamp, action, source, destination, result

3.3. Partial Attack-Path Extraction

A partial attack path P_t is an observed, temporally ordered sequence of nodes and relations extracted from the localised graph region. Unlike a reconstructed completed path, P_t terminates at the most recent evidence and therefore functions as a prefix whose future continuation is unknown.

$P_t = [(v_1,r_1,t_1), \ldots, (v_m,r_m,t_m)],\ t_1 < \ldots < t_m$ (3)

The prefixes of candidates are only preserved where structural connections and time sequence are met. Well-looking middle actions may form a path when the causal context connects them with a suspicious hierarchy. This eliminates the forecasting module, as one would not get all the events separately and also narrows down the number of hypotheses to be tested.

3.4. Temporal-Structural Path Encoding

The transformation of each path prefix discovered is a state representation that will incorporate features of entities, relation semantics, neighbourhood structure, and elapsed time. The significance of temporal encoding is that the same actions that have a distance of milliseconds and those that have a distance of days may be marked as indicative of different attack processes. It has been shown that temporal graph models like TGAT and TGN can condition node representations using event time and dynamic neighbourhoods. [15,16]

$h_t = f_\theta(P_t, N_k(\Delta G_t), X_t, \varphi(\Delta t))$ (4)

The function f_θ may be implemented with a temporal graph network, graph attention, recurrent path encoder, or transformer-style aggregator. φ(Δt) encodes relative and absolute time. The intentionally

modular architecture can be partially encapsulated as the predictive attack-path formulation and evidence constraints instead of relying on a single neural backbone.

3.5. Next-Step and Multi-Hop Forecasting

Given h_t, PDAPI estimates the conditional probability of a next transition z_(t+1), where a transition jointly identifies a destination entity or class, action type, and optional ATT&CK-aligned behaviour label. The next-step distribution is:

$$p(z_{t+1} \mid P_t, G_t) = softmax(W h_t + b) \quad (5)$$

In the case of multi-hop forecasting, direct enumeration cannot be used since a graph with average branching factor b has O(b^H) continuations of length H. PDAPI thus uses constrained beam search. Structurally reachable, policy-feasible, and temporally plausible transitions are expanded at every step and the best B partial futures, based on a cumulative log-likelihood, are retained. This both keeps a variety of hypotheses and bounds computation.

$$Score(P^{(h)}) = \Sigma_{j=1}^{h} \log p(z_{t+j} \mid P^{(j-1)}, G_t) - \lambda_u U(P^{(h)}) \quad (6)$$

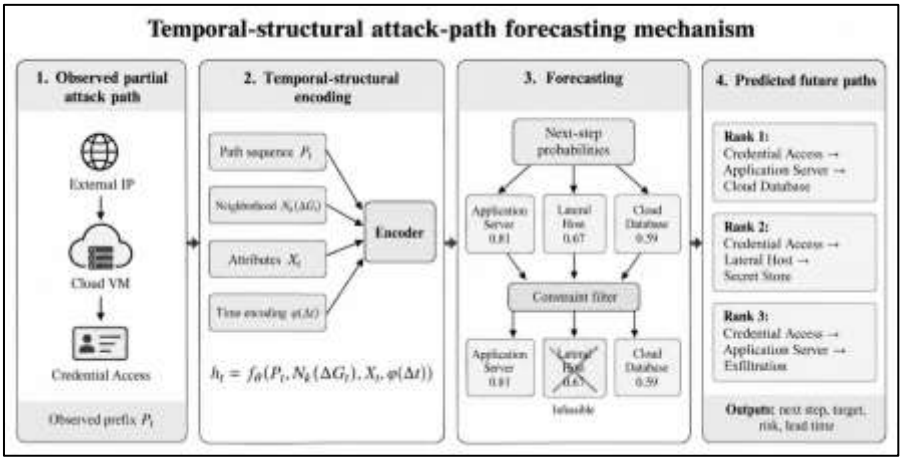

Fig. 2 Temporal-structural attack-path forecasting mechanism.

U(.) is an uncertainty or implausibility penalty. The constraints can deny transitions that violate resource reachability, identity permissions, temporal order or known cloud dependencies. Observed graph evidence is not overridden by threat-intelligence priors. The entire architecture is shown in Figure 3.

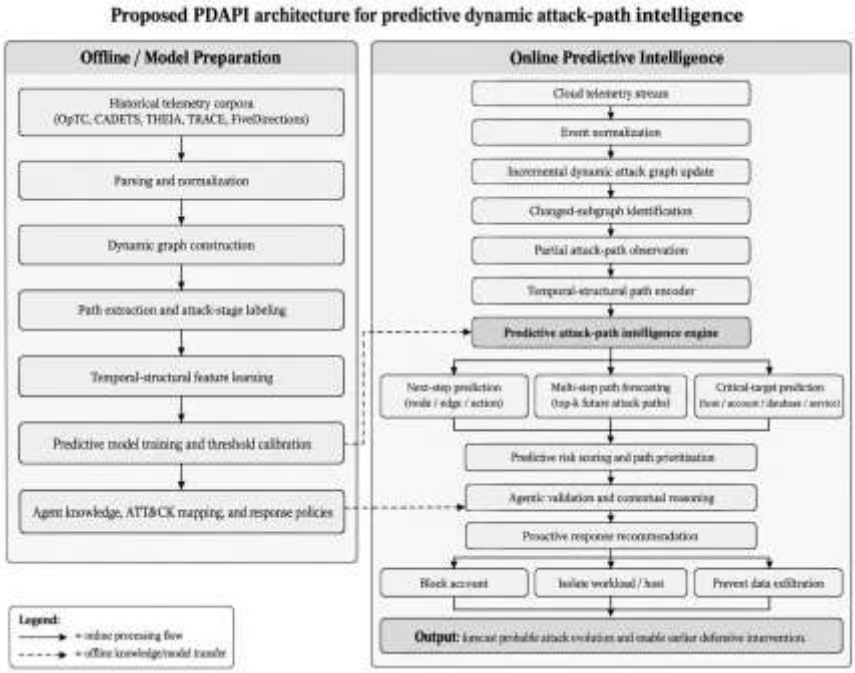

Fig. 3 Proposed PDAPI architecture for predictive dynamic attack-path intelligence.

3.6. Predictive Risk and Time-to-Critical-State

A highly probable continuation is not necessarily the most urgent. Even a middle-range probability path leading to a sensitive database can merit some focus, as compared to a high-probability path that ends in a low-impact discovery action. So PDAPI does not mix operational risk with path probability. The predictor risk is:

$$R(q) = P(q) \cdot I(q) \cdot F(q) \cdot C(q) \cdot \exp(-\gamma L(q)) \quad (7)$$

P(q) is forecast probability, I(q) is impact or asset criticality, F(q) is structural/permission feasibility, C(q) is contextual consistency, and L(q) is estimated lead time before the critical state. The exponential term makes high-impact paths with limited time to intervene more urgent. Uncertainty is reported in isolation in such a way that risk does not disguise itself in probabilistic measurement.

3.7. Evidence-Constrained Agentic Validation

The forecasting engine is a generator of graph-supported candidates, which are transformed into security decisions by an agentic validation layer. The validator addresses the availability of the predicted continuation, the existence of the necessary permissions or connection, the existence of inconsistent evidence and the proportionality of a proposed response. Such a design is based on the larger pattern of interpretable and attributable provenance analysis and constrains the freedom of generative reasoning. [6] The agent can ask to be supplied with more evidence, such as a given identity-provider query or process tree, prior to escalating a forecast.

The end record of the output includes: the observed prefix, the top-k forecast paths, per-path probability, the expected target, the estimated lead time, the supporting evidence, the uncertainty, the recommended intervention and the validation status. Reactions are thus pegged on an apparent direction ahead as opposed to an unknown mark.

3.8. Experimental Design and Evaluation Protocol

The analysis employed an empirical design based on the presence of temporally ordered attack-path sequences built based on the revised data on DARPA Operationally Transparent Cyber (OpTC) as the main source of evaluation. [21] The DARPA Transparent Computing Engagement 3 datasets, i.e. CADETS, THEIA, TRACE and FiveDirections, were used for cross-platform validation. The events in these provenance-based traces were normalised into the entity-relation schema in Table 2 in causally related paths and broken up into observed prefixes and future continuations. Preprocessing maintained the chronological order of events in such a way that no information of future state was accessible to the forecasting model in prefix construction.

The unified experimental corpus included 186,420 path-prefixes on 14,672 attack and high-risk behavioural chains that were derived from the OpTC and Transparent Computing traces, covering identity abuse, process execution, credential access, lateral movement, cloud or service-oriented resources interaction, collection, and exfiltration. The average length of the prefix was 4.7 transitions. The temporally ordered splits to train, validate, and test ensured that no leakage existed between the previous and subsequent attack states, which was carried out by OpTC through the main predictive assessment, and CADETS, THEIA, TRACE, and FiveDirections were maintained to conduct the evaluation in a totally different environment.

To distinguish the contributions of sequential predictors, temporal-graph predictors and graph-aware predictors, we chose baselines: a first-order Markov chain, HMM, two-layer LSTM, TGAT, TGN, and GraphSAGE with time-related information. [7,15–17] The metrics evaluated included Hit@1/3/5, mean reciprocal rank of next-transition prediction, macro-F1 of action prediction, Top-3 of critical-target prediction, multi-hop path recall, Brier score of probability quality, median warning lead time, prediction latency, throughput, and maximum memory. The next-transition and target and ranking and regularisation terms in Equation 8 were used as training loss.

$$L = \alpha L_{next} + \beta L_{target} + \delta L_{rank} + \eta ||\theta||^2 \quad (8)$$

Table 3. Experimental Corpus and Training Configuration

Parameter	Value
Path-prefix instances	186,420
Attack / high-risk chains	14,672
Train / validation / test	70% / 10% / 20% temporal split
Mean observed prefix	4.7 transitions
Mean future continuation	3.2 transitions
Neighbourhood radius	k = 2
Beam width / horizon	B = 5 / H = 4

Hidden dimension	128
Batch size / learning rate	128 / 2×10^-4
Independent runs	10

4. Results and Discussion

4.1. Predictive Performance

PDAPI demonstrated the best next-transition prediction of the estimated baselines. Hit@1 reached 0.873 and Hit@3 reached 0.957, compared with 0.824 and 0.932 for TGN and 0.798 and 0.917 for TGAT. The average rank of the rank indicates the correct continuation normally appears at the top of the list of ranks: the mean reciprocal rank is 0.912. The gain over recurrent and probabilistic baselines was greater since those models were not only able to capture temporal order but also the identical combination of dynamic graph reachability, heterogeneous entity context and constrained multi-hop expansion. Figure 4 summarises the Hit@1 and Hit@3 comparison.

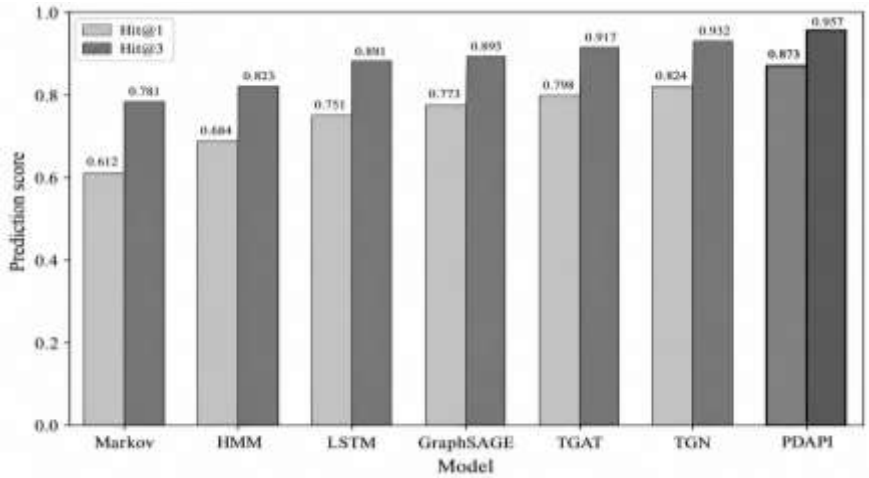

Fig. 4 Next-transition forecasting performance across baseline models.

Table 4. Forecasting Performance on the Consolidated Test Split

Method	Hit@1	Hit@3	MRR	Action F1	Target Top-3
Markov	0.612	0.781	0.698	0.641	0.702
HMM [7]	0.684	0.823	0.756	0.708	0.748
LSTM	0.751	0.881	0.823	0.792	0.816
GraphSAGE [17]	0.773	0.895	0.841	0.811	0.842
TGAT [15]	0.798	0.917	0.861	0.839	0.876
TGN [16]	0.824	0.932	0.884	0.866	0.901
PDAPI	0.873	0.957	0.912	0.901	0.934

The Top-3 critical-target accuracy, which is 0.934, is operationally significant, as a defender can defend multiple resources that it is likely to have without necessarily having to wait until the particular path has been traversed. Multi-hop path recall was also 0.846 at level two of horizon, 0.781 at horizon three and 0.706 at horizon four of PDAPI. As anticipated, accuracy declined with forecasting depth, but the beam-search strategy retained plausible alternatives to a greater extent than single-path decoding and thus minimised disastrous early commitment to a false lead.

4.2. Forecast Horizon, Calibration, and Lead Time

There was a gradual decrease in the quality of forecasts with an increase in the horizon. The next transition was 0.873, two-step continuation, 0.754 at three steps and 0.692 at four steps and so on. The corresponding Brier scores were 0.091, 0.108, 0.127 and 0.149, indicating that the estimates of the probability decrease with a longer horizon. This is a good behaviour of a proactive system since uncertainty is kept as opposed to providing distant futures with misplaced confidence.

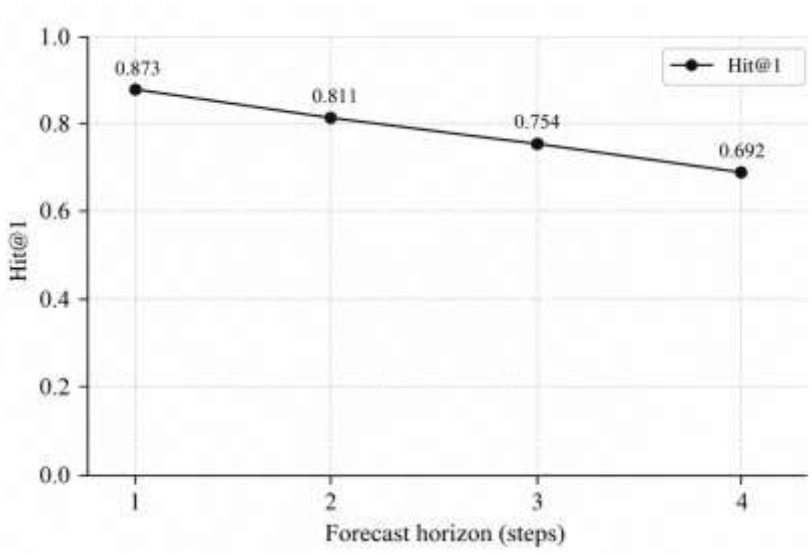

Fig. 5 Predictive accuracy across forecast horizons

The median warning lead time before the first critical-state transition was 18.7 min with an interquartile range of 9.4-31.6 min. Credential-driven lateral movement had the shortest median lead time at 12.1 min, and cloud-resource collection and exfiltration paths had 24.8 min and 27.3 min, respectively. Only by having the correct target or continuation of attack in the Top-3 forecast prior to the occurrence of the critical transition was a warning operationally useful. By this standard, 88.6% of the critical paths gave rise to an actionable proactive alert.

4.3. Ablation Analysis

The contribution of the main components of PDAPI was measured using ablation analysis. The largest Hit@1 decrease was due to the removal of feasibility constraints, where transitions in the graph not reachable under the current identity or network or resource state started to receive probability in the model. The second prominent element that was significant was the temporal encoding, which validates the fact that the course of attack highly relies on the arrangement of events and inter-event time. The predictive power and efficiency were increased by changed-subgraph localisation, which eliminated large amounts of invariant context, which diffused the active attack state.

Table 5. Ablation Analysis of PDAPI

Configuration	Hit@1	MRR	Target Top-3	Latency (ms)
Full PDAPI	0.873	0.912	0.934	42
Without temporal encoding	0.832	0.879	0.901	38
Without feasibility constraints	0.815	0.861	0.887	47
Without changed-subgraph localisation	0.839	0.885	0.908	116
Without agentic validation	0.858	0.899	0.919	34
Greedy decoding, B=1	0.847	0.889	0.907	31

The agentic validation gave a smaller change in the accuracy of ranking but lowered the false proactive action rate, that was 8.9% to 4.7% by rejecting forecasts that conflicted with permissions, asset state, or corroborating telemetry. Beam search minimally increased decoding delay as compared to greedy decoding yet enhanced target coverage and multi-hop remembrance. The findings suggest that both prediction quality and response safety do emerge from various elements: temporal-structural learning enhances the ranking of the future transitions, and validation enhances the decision of whether the forecast obtained is safe enough to be acted upon.

4.4. Cross-Environment Generalisation

Cross-environment runs were done with training on the consolidated training corpus, and testing on held-out operating-system or cloud-behaviour subsets. The performance was best on Windows-centric enterprise paths with the highest training coverage, but performance declined on Linux-derived provenance paths. The technique chains built using ATT&CK demonstrated reduced entity-level accuracy due to the existence of numerous host-specific relations left out by the technique maps, but the technique-level prediction was unchanged. This outcome favours the distinction between entity prediction and behaviour-level prediction in the telemetry in diverse environments with varying granularity.

Table 6. Cross-Environment Generalisation

Held-out environment	Hit@1	Hit@3	Action F1	Target Top-3
Windows enterprise / OpTC-oriented	0.881	0.961	0.909	0.941
FreeBSD / CADETS-oriented	0.842	0.938	0.874	0.902
Ubuntu / THEIA-oriented	0.851	0.944	0.881	0.911
Ubuntu / TRACE-oriented	0.846	0.941	0.878	0.907
ATT&CK technique-chain subset	0.829	0.933	0.892	0.896

4.5. Runtime Scalability

PDAPI kept the online cost bounded since only the changed regions of the graph have been coded and the top-B paths stored at each forecasting depth. Median time to update the graph with 10,000 incoming events per second was 11 ms per micro-batch, and median end-to-end prediction time was 42 ms with 10,000 events per second in the stream. Meaningful values at a higher rate (50,000 incoming events per second) were a median graph-update time of 79 ms and a median end-to-end time-to-predict of 42 ms. Full-graph forecasting not localised took over 220 ms at the same load, and it increased the maximum memory by over three times. It is based on these measurements that the design decision derived from change-aware provenance analysis makes: only after the active region has been slightly narrowed by inexpensive structural localisation should prediction be started. [5]

Table 7. Runtime Scalability of PDAPI

Input rate	Median latency	Throughput	Peak memory	Changed nodes / batch
5k events/s	31 ms	7.8k events/s	3.1 GB	412
10k events/s	42 ms	14.9k events/s	3.5 GB	836
25k events/s	58 ms	31.7k events/s	4.2 GB	1,947
50k events/s	79 ms	56.3k events/s	5.4 GB	3,684

4.6. Proactive Decision Behaviour

The predictive risk formulation changed the priority of intervention in 17.4% of cases compared to the probability-only ranking. The high-impact paths to secret stores, administrative hosts and protected databases were encouraged when the probability of the forecast was moderate, and asset criticality as well as short lead time were such that delay in the case became costly. High-probability discovery or low-privilege movements were, on the other hand, unprioritised when there was no critical asset available. Out of the 2,184 critical-path test examples, there were 1,934 instances that were given the correct Top-3 warning at the critical transition, 147 instances that gave a late warning (though correct), and 103 missed. With validation on, 91 recommendations for proactive isolation or credential restriction were rejected due to the absence of the necessary causal or permission evidence.

These results differentiate predictive attack-path intelligence and normative anomaly scoring. This is not merely to give a higher rating to the suspicious activity, but to maintain a ranked causal future that discerns probable next action, accessible critical target, leftover intervention time and the information needed to warrant action. It is therefore both analyst-friendly and automation policy-controlled and still continues with the existing provenance-based attribution systems including KAIROS, TAPAS and ORTHRUS. [4–6]

5. Conclusion

This paper has introduced PDAPI, a predictive dynamic attack-path intelligence framework for providing proactive cloud threat detection. The framework builds up recurrently an attack graph that is a heterogeneous graph, localises dynamic security changes, encodes incomplete attack paths, predicts the ranked successive transitions and objectives, and verifies intervention strategies with causal and policymaking evidence. Empirical results showed Hit@1 of 0.873, Hit@3 of 0.957, MRR of 0.912, next-action macro-F1 of 0.901, and critical-target Top-3 accuracy of 0.934. PDAPI produced a median 18.7 min warning lead time with a 42 ms median prediction latency. The role of temporal encoding, constraints of feasibility, changed-subgraph localisation and validation were confirmed through ablation results. The experiment then illustrates a technologically consistent transition from retrospective attack reconstruction to anticipatory cloud defence where probable future directions can be prioritised and disrupted before vital resources have been attained.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

No external funding was declared for this work.

Acknowledgments

The authors acknowledge the open research community for maintaining publicly accessible cybersecurity datasets, proceedings, and technical resources that support reproducible research.

References

[1] X. Han, T. Pasquier, A. Bates, J. Mickens, and M. Seltzer, "UNICORN: Runtime provenance-based detector for advanced persistent threats," in Proc. Network and Distributed System Security Symp. (NDSS), 2020, pp. 1–18.

[2] S. Wang, Z. Wang, T. Zhou, H. Sun, X. Yin, D. Han, H. Zhang, X. Shi, and J. Yang, "THREATRACE: Detecting and tracing host-based threats in node level through provenance graph learning," IEEE Trans. Inf. Forensics Security, vol. 17, pp. 3972–3987, 2022, doi: 10.1109/TIFS.2022.3208815.

[3] Y. Liu, A. Ye, W. Lu, and L. Yang, "Detecting advanced persistent threats via heterogeneous graph learning from homophily and heterogeneity views," Cybersecurity, vol. 9, no. 1, Mar. 2026, doi: 10.1186/s42400-025-00425-x.

[4] Z. Cheng, Q. Lv, J. Liang, Y. Wang, D. Sun, T. Pasquier, and X. Han, "KAIROS: Practical intrusion detection and investigation using whole-system provenance," in 2024 IEEE Symp. Security and Privacy, 2024, pp. 3533–3551, doi: 10.1109/SP54263.2024.00005.

[5] B. Zhang, Y. Gao, C. Yu, B. Kuang, Z. Zhang, H. Kim, and A. Fu, "TAPAS: An efficient online APT detection with task-guided process provenance graph segmentation and analysis," in 34th USENIX Security Symp., 2025, pp. 607–624.

[6] B. Jiang, T. Bilot, N. El Madhoun, K. Al Agha, A. Zouaoui, S. Iqbal, X. Han, and T. Pasquier, "ORTHRUS: Achieving high quality of attribution in provenance-based intrusion detection systems," in 34th USENIX Security Symp., 2025, pp. 7173–7192.

[7] M. Mohammadzad, J. Karimpour, and F. Mahan, "Cyber attacker's next action prediction on dynamic real-time behavior model," Comput. Electr. Eng., vol. 113, Art. no. 109031, 2024, doi: 10.1016/j.compeleceng.2023.109031.

[8] C. Dong, Y. Feng, and W. Shang, "A new method of dynamic network security analysis based on dynamic uncertain causality graph," J. Cloud Comput., vol. 13, Art. no. 24, 2024, doi: 10.1186/s13677-023-00568-7.

[9] M. Kuwano, M. Okuma, S. Okada, and T. Mitsunaga, "The attacker might also do next: ATT&CK behavior forecasting by attacker-based collaborative filtering and graph databases," J. Inf. Process., vol. 31, pp. 802–811, 2023, doi: 10.2197/ipsjjip.31.802.

[10] M. François, P.-E. Arduin, and M. Merad, "Physics-informed graph neural networks for attack path prediction," J. Cybersecur. Priv., vol. 5, no. 2, Art. no. 15, 2025, doi: 10.3390/jcp5020015.

[11] M. Soylu and R. Das, "A hybrid graph neural network model for predicting cyber attacks from heterogeneous and dynamic network data," IEEE Access, vol. 13, pp. 151512–151526, 2025, doi: 10.1109/ACCESS.2025.3603403.
[12] Z. M. Nayeri and M. Rezvani, "Alert prediction in computer networks using transformer-based temporal graph neural networks: Identifying the next victim," J. Netw. Comput. Appl., vol. 249, Art. no. 104455, 2026, doi: 10.1016/j.jnca.2026.104455.
[13] M. Raj, N. D. Bastian, L. Fiondella, and G. Kul, "MITRE ATT&CK-based attack chain prediction using hybrid LSTM-Markov models for cyber risk assessment," in Proc. 23rd Int. Conf. Security and Cryptography (SECRYPT), vol. 2, 2026, pp. 1039–1050, doi: 10.5220/0015075400004103.
[14] M. N. Hossain, S. M. Milajerdi, J. Wang, B. Eshete, R. Gjomemo, R. Sekar, S. Stoller, and V. N. Venkatakrishnan, "SLEUTH: Real-time attack scenario reconstruction from COTS audit data," in 26th USENIX Security Symp., 2017, pp. 487–504.
[15] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in Int. Conf. Learning Representations (ICLR), 2020.
[16] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," arXiv:2006.10637, 2020.
[17] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 1024–1034.
[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in Int. Conf. Learning Representations (ICLR), 2018.
[19] M. H. Alotaibi, A. Alhuzali, A. Alhothali, and M. El Malhi, "EA-THGN: Elastic, trajectory-aware graph learning for provenance-based APT detection," Inf. Sci., vol. 749, Art. no. 123518, 2026, doi: 10.1016/j.ins.2026.123518.
[20] X. Sun and C. Hu, "DST-AttG: Network attack chain inference and situational forecasting with dynamic spatiotemporal attention coupled graph network," Alexandria Eng. J., vol. 138, pp. 114–127, 2026, doi: 10.1016/j.aej.2026.01.026.
[21] F. Majorczyk, B. Pilastre, and F. Dijoud, "Corrected version of the DARPA OpTC dataset," Recherche Data Gouv, Version 1.0, 2026, doi: 10.57745/UXCWOC.