

Reproducibility and Generalisability of Provenance-Based Intrusion Detection Systems on Real SOC Data

Benjamin Irvine

Computer Science and Software Engineering

University of Canterbury

Christchurch, New Zealand

bir17@uclive.ac.nz

Abstract—Provenance-based intrusion detection systems (PIDS) report near-perfect detection results, almost exclusively on curated DARPA benchmarks and under their authors’ own evaluation conditions. This project asks whether those results can be reproduced, and whether they transfer to the data a real Security Operations Centre produces, using one standardised framework, PIDSMaker. For reproducibility, NodLink’s published result (precision 0.14 at full recall) was evaluated across ten seeded runs, a full threshold analysis, and an inspection of its implementations. The result does not reproduce: two independent reproductions agree at precision below 0.005, the claimed operating point is unreachable at any threshold, and the causes trace to documented divergences between the paper, its released code, and the framework’s reduced reimplementation. For generalisability, four screened systems were carried onto the Precinct 6 dataset, operational telemetry published openly by WitFoo, in a natural experiment: 54 labelled attacks inside real enterprise days, evaluated per incident at three data densities. No system offers a usable operating point as configured, which system is best depends entirely on the alarm budget, and behaviour is determined less by architecture than by three things papers treat as fixed: threshold rules, data density, and the match between each system’s inductive bet and the attacks’ shape. The report closes with evaluation recommendations: report ranking quality and the cost of recall separately from threshold rules, report final training epochs with labelled oracle bounds, and treat ranking metrics over tied scores with caution.

Index Terms—intrusion detection, provenance graphs, reproducibility, security operations

I. INTRODUCTION

Provenance-based intrusion detection systems (PIDS) promise to catch the advanced, multi-stage attacks that signature-based defences miss, and recent systems report near-perfect results. Those results, however, are produced by the systems’ own authors, under evaluation conditions that are often not fully released, and almost exclusively on one family of curated benchmarks, the DARPA Transparent Computing datasets [1], [2]. A systematic reproduction attempt found none of eight such systems fully reproducible end-to-end [1], and evaluations on real industrial data report failure modes the benchmarks never surface [3]. Two questions follow: whether the published numbers can be reproduced at all, and whether

they mean anything for data a real Security Operations Centre produces.

This project addresses both questions through one standardised framework, PIDSMaker [4]. For reproducibility (RQ1), it evaluates NodLink [5], a recent system reporting precision 0.14 at perfect recall, against its published claims. For generalisability (RQ2), it carries four screened systems onto the Precinct 6 dataset [6], operational telemetry published openly by WitFoo, in a natural experiment: real attacks embedded in real enterprise days.

The findings are unfavourable to the published numbers but constructive about why. NodLink’s published result does not reproduce: two independent reproductions agree two orders of magnitude below the paper’s precision, and the claimed operating point is unreachable at any threshold. On operational data, no system offers a usable operating point as configured; which system is best depends entirely on the alarm budget; and the dominant causes of failure are not the models but the threshold rules layered on them, together with data density and the match between each system’s inductive bet and the attacks’ shape.

This project makes five contributions:

- a documented reproduction study of NodLink on CADETS_E3, quantifying run-to-run (73% relative standard deviation) and inference-time variance, demonstrating the published operating point unreachable at any threshold, and tracing the gap to documented implementation divergences;
- a reproducibility tool: per-epoch checkpointing and re-seeded re-evaluation on the PIDSMaker fork, so a trained state can be handed over and verified rather than re-trained.
- a translation pipeline from WitFoo’s operational export into PIDSMaker, with three label mappings, per-incident ground truth, deterministic density sampling, and constant-memory streaming.
- a four-system evaluation on that data at three densities with per-incident resolution, an incident characterisation linking attack shape to detectability, and a conventional-ML baseline bounding the results from below;

- evidence-backed evaluation practices: report ranking quality and the cost of recall separately from the threshold rule, report the final training epoch with a clearly labelled oracle bound rather than the best test-set epoch, and treat ranking metrics over tied scores with caution.

II. BACKGROUND AND OBJECTIVES

This project is an individual final-year research project in the Department of Computer Science and Software Engineering at the University of Canterbury, supervised by Etienne Borde, Mengmeng Ge, and Maria Kainat. The industry data central to the project’s generalisability question is provided by WitFoo, Inc., whose co-founder Charles Herring engaged directly with questions raised about the data; this exchange led WitFoo to correct and regenerate the dataset itself (Section IV-B).

Attacks vary in how visible they are: some announce themselves in a single detectable action, while advanced persistent threats unfold as long sequences of individually unremarkable actions. Provenance-based intrusion detection systems (PIDS) target this second kind. They take in provenance graphs, where nodes are entities such as processes, files, and network endpoints and edges are the interactions between them, and apply different techniques to score how anomalous the activity is.

In operational deployments, a PIDS is not the first line of detection. Signature-based detection remains the mainstay of operational security monitoring [7], and commercial endpoint detection tools generate their alerts by matching rules over known attack behaviours, with provenance analysis layered downstream of those alerts [8]. Provenance-based systems therefore target the advanced persistent threats that evade these upstream stages [3], [9]. Two consequences follow for this project. First, the field’s two evaluation priorities, recall (the fraction of attack activity flagged) and precision (the fraction of alarms that are real), trade off as risk against benefit: a missed attack at this last line of detection is the risk a deployment carries, while the benefit is the attacks the system catches, and false alarms erode that benefit by consuming the triage capacity needed to act on it [2]. If the benefit is small enough, a deployment may rationally prefer to carry the risk and not rely on the system at all. The two priorities are also coupled: low precision degrades realised recall, because alarms beyond an analyst team’s triage capacity are never analysed. Second, a dataset is a fair test of a PIDS only to the extent that its attacks are of the kind that would reach one (Section V-B).

PIDSMaker is a standardised framework implementing eight provenance-based detection systems in one pipeline [4]. Standardising ingestion, featurisation, and evaluation means that performance differences are attributable to the systems themselves rather than to differences in plumbing. The framework also adopts ADP (Attack Detection Precision), a threshold-independent measure of how well a system ranks attack entities above benign ones [2]. No prior UC project precedes this work; the pre-existing results it builds on are the systems’ published evaluations and PIDSMaker itself.

The industry data for RQ2 is the Precinct 6 Cybersecurity Dataset [6]: a large-scale labelled dataset derived from production Security Operations Centre telemetry, anonymised and published openly by WitFoo. The telemetry is processed by Precinct, WitFoo’s security-operations platform, and the labels record the state of Precinct’s automated correlation engine rather than analyst verdicts. Unlike curated research datasets, it carries the volume and irregularities of operational data.

Three constraints shaped the project. All experiments run on UC’s shared research compute cluster (SLURM-scheduled GPU nodes with per-node memory ceilings of roughly 190–290 GB), which bounds the data volumes any system can be tested at. PIDSMaker, recommended by the supervision team, was selected as the harness, fixing the candidate systems to its eight implementations. And the 300-hour individual project budget bounds the number of systems and datasets that can be studied. Within those constraints, four systems were selected for full study: NodLink (the original subject of study), ORTHRUS, Velox, and R-CAID, chosen by an experimental screening of all eight implementations described in Section IV-B (Table I).

This project asks two questions. RQ1: Can the published results of provenance-based IDS be reproduced within a common framework, and what does reproduction depend on? RQ2: To what degree do these systems’ results hold on real industry data, and what determines their behaviour there?

Answering RQ1 matters because published results are currently taken on trust; answering RQ2 matters because the decision to deploy a PIDS rests on whether benchmark results transfer to operational data, and if they do not, which properties of the data drive the difference. For RQ1, success means reproducing the metrics reported by the original papers within the common framework, quantifying run-to-run variance, since a single-run comparison is uninterpretable when variance is large, and documenting every configuration choice the results depend on that the papers do not state. For RQ2, success means producing the same measurements on the industry data for all four systems, then descending to individual incidents, because aggregate metrics can hide incident-level behaviour. Throughout, evaluation uses the framework’s standard metrics: precision, recall, F1, and false-discovery rate over flagged entities, with ADP for threshold-independent ranking quality, plus the per-incident detection coverage introduced in Section IV-B.

III. RELATED WORK

The literature was assembled in three ways. The supervisory team seeded it at project definition with the NodLink paper [5], the reproducibility study of Abrar et al. [1], and the companion project’s KnowHow paper [10] for context. Targeted Google Scholar searches located the survey and the industrial-perspective literature, and the remaining system papers were snowballed from the citations of the PIDSMaker framework [4]. The depth of review follows the study’s scope: the eight implemented systems’ papers, the surveys, and the industrial-perspective evaluations were reviewed in depth,

while systems outside the framework, such as KnowHow, were read for context only.

Inam et al. systematize the provenance-based auditing and detection field [11]. Within it, the four systems under study each make a distinct bet. NodLink [5] targets online detection under time and resource constraints, modelled as a Steiner-tree problem, and scores by variational autoencoder (VAE) reconstruction: anomaly is what the model cannot reconstruct. ORTHRUS [12] bets on temporal dynamics: a temporal graph network predicts interaction types, and anomaly is an interaction the learned dynamics find surprising; its published design thresholds on validation scores with a clustering step. Velox [2] is the lightweight variant of the same objective family: a linear encoder and MLP predicting edge types, with no temporal machinery. R-CAID [13] bets on root-cause identity, restructuring the graph to connect every node to its provenance roots on the premise that where a chain started is the discriminative signal.

Abrar et al. [1] attempted to reproduce eight deep-learning PIDS and catalogued the obstacles: non-functional code releases, missing documentation, unavailable datasets, unclear procedures. This work is the constructive complement: reproduction within a common framework, asking what the results depend on. The evasion literature [14], [15] critiques robustness from the attacker’s side. It asks a different question from this project, but its core observation, that attack-like and benign behaviour separate only fragilely, is one this work rediscovers from the defender’s side on operational data.

Dong et al. [9] surveyed practitioners and identified three gaps between academia and deployment: client-side overhead, imbalanced alarm-triage cost, and excessive server-side memory consumption; this project measures the second and third directly. Xiao et al. [3] ran five PIDS on a technology company’s internal data and found poor cross-host portability, weak detection of real-world attacks, and high false-positive rates against evolving benign activity: the closest published relative to RQ2. This evaluation differs in using a publicly released operational dataset, per-incident ground truth, and one common framework, making the same question answerable by others. The synthesis literature attacks the same root problem from the other direction: ProvSyn [16] generates synthetic provenance graphs to make training data more representative. This work instead brings the systems to real data.

IV. PROPOSED SOLUTION

A. Design and Implementation

The translation pipeline converts WitFoo’s export into PIDSMaker’s input format: node and edge files go in, and the pipeline filters to a single organisation, applies a graph mapping, and emits the node tables, event table, and ground-truth list PIDSMaker ingests, with each day’s role (training, validation, testing) declared in configuration.

A mapping is needed because of a mismatch: in the WitFoo data the events (edges) carry the malicious labels, while the systems under test judge nodes. Three mappings were designed: A marks the nodes at either end of a malicious edge,

B creates an intermediary node for each event carrying the event’s text and label, and C is a variant of B whose connecting edges carry the event’s action type. Mapping B was selected: it preserves the event’s descriptive text for featurisation, gives ground truth at the granularity of individual events, and does not leak the answer through graph structure, because the intermediary node is created for every event, benign and malicious alike, so the topology is uniform and only the withheld labels differ. Mappings A and C were compared with B in theory only; their empirical evaluation was descope by supervisor decision to prioritise scale testing.

Ground truth comes from the dataset’s own labels: events that Precinct’s correlation engine attached to an incident and, within those, the tier the engine actioned (“Disrupted”). No analyst verification exists in the dataset, so this is an operational ground truth. A per-incident emitter partitions this ground truth into one file per attack incident, enabling incident-level evaluation. WitFoo’s data arrived in two successive exports covering the same underlying telemetry, yet they disagreed on the labelling of 22 events, so ground truth is regenerated from each export’s own labels rather than reused.

Two experiment designs were used in sequence, one per export of the dataset. In the first export, the organisation’s attacks existed only in a multi-year archive with no overlapping benign capture, so benign traffic was time-shifted to overlay the attack period, an artificial construction retained as a baseline. The second export, regenerated by WitFoo after the quality findings this project reported upstream (Section IV-B), contained an organisation whose attacks occur naturally inside its live benign traffic, enabling a natural experiment with no manipulation. A third design, promoting former attack days into training with their attacks stripped, was tested and rejected: it destabilised one system’s training.

For RQ1, checkpointing and reseeding capabilities were added to the PIDSMaker fork, saving model weights after every epoch. Their purpose is reproducibility as hand-off: randomness in training can prevent others from regenerating a result even with the correct configuration and seeds, so a checkpoint lets the trained state itself be passed on and verified rather than recreated.

Finally, a deterministic sampling option was added to the translator: retaining one in K benign events by hash while always keeping every attack event, enabling dataset density to be varied controllably for memory-ceiling and cause-isolation experiments, with ground truth provably unchanged. To support this at full scale the translator processes the export as a stream, reading the sharded input piece by piece, so memory use stays constant regardless of export size.

B. Method

System selection. All eight framework implementations were screened with an exploratory run on the DARPA CADETS_E3 benchmark before any industry-data work, using threshold-independent ranking quality as the criterion: systems with mean ADP ≤ 0.1 in both this study’s screening and the published multi-run results of Bilot et al. [2] were set aside,

TABLE I
SCREENING OF ALL EIGHT FRAMEWORK IMPLEMENTATIONS ON
CADETS_E3: ONE 50-EPOCH EXPLORATORY RUN PER SYSTEM (THIS
STUDY) AGAINST THE PUBLISHED FIVE-RUN RESULTS OF BILOT ET
AL. [2].

System	This study		Bilot et al.		Verdict
	Best	Mean	Best	Mean	
NodLink	0.899	0.136	0.96	0.33	kept
ORTHROS	0.852	0.338	1.00	0.94	kept
Velox	0.706	0.309	1.00	0.94	kept
R-CAID	0.700	0.254	0.44	0.21	kept
Flash	0.379	0.066	0.34	0.10	set aside
KAIROS	0.431	0.035	0.01	0.01	set aside
MAGIC	0.020	0.015	0.06	0.03	set aside
ThreaTrace	0.011	0.009	0.01	0.01	set aside

the cross-check guarding against configuration error. Four systems (ThreaTrace, MAGIC, KAIROS, Flash) failed the screen; MAGIC additionally computes its detection threshold from test data, precluding a deployment-style evaluation. The remaining four were studied in full (Table I).

Evaluation pipeline. Each run proceeds through six stages: graph construction from the event tables, optional transformation (e.g. R-CAID’s pseudo-graphs), featurisation of each node’s textual attributes into embeddings, batching for the GPU, training on the training days’ “normal”, and evaluation, which scores the test day’s entities, applies the system’s threshold rule, and compares the flagged set against ground truth. Each system builds its graphs and computes its threshold differently.

Reproducibility protocol. Four things were fixed and recorded so that any result can be regenerated exactly: dataset revisions (each export referenced by its immutable revision hash, so later dataset rebuilds cannot silently change inputs); random seeds, fixed in configuration for every run; the complete configuration of every experiment, version-controlled on the project fork; and per-epoch model checkpoints of Section IV-A. Ground truth files are likewise generated by committed scripts and stored in the repository.

Result provenance. Every reported number is computed from the framework’s saved evaluation artifacts (per-epoch score files) rather than from run logs. The framework names artifact directories by configuration hash, so an artifact’s identity cannot be inferred from file dates or log order. Extraction therefore maps each hash to its system and dataset via the `scores_file` lines of that run’s own scheduler log, and validates every artifact on load against the dataset’s known ground-truth counts (3,062 attack entities at 2M; 3,040 on the second export), refusing a mismatch. The incident-level metric computation replicates the framework’s own code and was unit-tested to bit-identical agreement before use.

Smoke-then-scale. No dataset was committed to compute before being measured. Each new export was first censused (per-organisation, per-day, per-label event counts; incident structure; timestamp integrity) to establish which days were attack-free, and where benign and malicious activity overlap.

Runs then started at the smallest useful scale and grew: a configuration was proven on the 2M export before any full-export attempt, and full-export attempts proceeded stage-by-stage so that each failure identified one bounded problem (a memory ceiling, a ground-truth mismatch) rather than a compound one. Three dataset scales recur throughout: the first export (2M events), and the second export (101M events at the pinned revision, of which the studied organisation’s test day holds 12M) sampled at one-in-eight (D8) and one-in-four (D4) benign retention, with every attack event always kept (Section IV-A).

Controlled comparisons. Experiments that compared configurations changed one variable at a time. The training-data enrichment experiment shows the practice: the baseline trained on the two naturally attack-free days (~99k events), and the enriched run changed exactly one thing, adding two former attack days with their malicious events removed (~420k events). When this destabilised one system, a further run changed one variable back: the same four-day content, deterministically sampled down to the baseline’s volume. The collapse persisted, attributing the effect to the content of ex-attack days rather than to data volume.

Metrics. Results report precision, recall, F1, and false-positive rate, plus two measures suited to operational data: the false discovery rate (FDR: the fraction of raised alarms that are wrong), because at operational scale a tiny false-positive rate (FPR: the fraction of benign entities flagged) can coexist with most alarms being false, and FDR is what an analyst triaging the queue experiences; and per-incident coverage (how many of the 54 incidents had an entity flagged), because aggregate event counts can hide whole incidents being missed. PIDSMaker’s own headline metric, ADP (Attack Detection Precision) [2], is also reported for the system screening and wherever ranking quality is compared across epochs or systems: a threshold-free measure of whether attacks can be found at high precision, computed as the area under the curve of attacks-detected against precision over the score ranking. Its sensitivity to how ground truth is grouped is examined in the Discussion.

Baseline. To calibrate the PIDS results, a conventional anomaly detector with no graph and no deep learning was run on the same experiment: an Isolation Forest [17] (scikit-learn [18]) over five per-event features. The event records offer six usable fields; three are excluded: the incident fields are the ground truth itself, the signature field is empty for every one of the test day’s 12.0M events, and event type nearly reproduces the labels (every attack event carries the single type EVENT, shared by only 0.4% of benign events). The five features are the standard signals a SIEM-style analytic would derive from the remaining host fields and timestamp: source-host rarity, destination-host rarity, host-pair rarity, burst rate (events in the same second), and time of day, with all rarity counts computed from the two clean training days only and unseen hosts flagged explicitly. The baseline’s role is a floor, not a competitor: a richer feature set could only raise the floor, making the comparison conservative in the PIDS’s favour. One mismatch

is inherent and stated wherever the comparison appears: the baseline flags events out of the 12M-event day while the PIDS flag nodes out of the ~ 2 M-node graph built from the 1-in-8 sampled day, so comparisons are made at equal alarm counts and only large differences are read as meaningful.

Uncertainty. Precision and recall carry 95% confidence intervals from bootstrap resampling of the scored population (20 resamples, with interval widths checked for stability). For NodLink, whose VAE samples at inference time, a fixed checkpoint was additionally re-evaluated under ten evaluation seeds to separate inference-time from training-time variance.

Model selection. PIDSMaker’s default reporting selects the epoch with the best test-set ADP; a form of test-set leakage, since the reported model is chosen for its performance on the evaluation data. Headline results here instead report the final training epoch throughout: a rule chosen without reference to test-day performance. Where training instability makes the final epoch unrepresentative, analysis sections additionally reference the epoch with maximum ADP, always labelled as an oracle upper bound, because the attack-free validation day offers no legitimate signal for epoch selection; the gap between oracle and final epochs is reported as a per-system stability measure. Tables additionally report the across-epoch mean $\pm$ s.d., so any single epoch can be read against the run’s spread (Table IV). The framework’s default is noted as a finding in its own right.

Feedback integration. Supervision ran as a weekly loop: results presented alongside a proposed direction, with the supervisors correcting course, such as redirecting the main thrust to the WitFoo dataset and requiring the training-collapse mechanism to be diagnosed before final runs. The collaboration with WitFoo ran by correspondence: dataset anomalies found during testing were reported upstream, and several were confirmed as real defects ($\sim 15\%$ duplicate incidents among them) and fixed in a subsequent dataset release.

V. EVALUATION AND DISCUSSION

A. Evaluation

1) *RQ1: Reproducing NodLink on CADETS_E3:* Across ten runs under the framework’s published NodLink configuration, mean precision was 0.0010 against the paper’s claimed 0.14, two orders of magnitude lower, with mean recall 0.825 against the claimed 1.0 (Table II). The published reproduction of Bilot et al. [2] sits in the same regime (precision < 0.005 over five runs), so the gap is not particular to this study’s setup.

A threshold sweep over the saved score files shows the gap is structural rather than a calibration problem. The test day holds 68 attack nodes against 281,517 benign; the attack nodes’ mean anomaly score (1.18) sits well above the benign mean (0.16), but 2.6% of benign nodes score above the attack mean and both distributions extend to the same maximum, so no threshold separates them cleanly. The best operating point found at the strongest epoch reaches precision 0.375 only at recall 0.088 (six true positives, all three campaigns touched); no threshold on these scores achieves precision 0.14 and recall 1.0 simultaneously.

TABLE II
RQ1: NODLINK ON CADETS_E3. TOP: THIS STUDY’S TEN PIDSMAKER RUNS (EPOCHS 0–11, PER-RUN BEST VALUES, FOLLOWING BILOT ET AL.) AGAINST THE PUBLISHED REPRODUCTION AND THE PAPER’S CLAIM. BOTTOM: THE SAME TEN RUNS EVALUATED UNDER THREE THRESHOLD RULES THE FRAMEWORK OFFERS.

	Li et al. [5] (paper claim)	Bilot et al. [2] (5 runs)	This study (10 runs)
ADP mean	—	0.33	0.265
ADP min / best	—	0.15 / 0.96	0.038 / 0.570
ADP RSD	—	56%	73%
Precision (mean)	0.14	< 0.005	0.0010
Recall (mean)	1.00	—	0.825
TP / FP at best precision	—	50 / 55,000	53 / 44,889

	NodLink rule (Grubbs 90th)	MAGIC rule (test mean)	max-val-loss (val max)
ADP mean	0.265	0.165	0.253
Precision mean	0.0010	0.0005	0.131
Recall mean	0.825	0.985	0.018
TP mean	53.6	56.5	0.8
FP mean	51,604	112,327	12.6

The reproduced numbers are themselves unstable. Under a fixed configuration differing only in seed, run-level ADP varies with a relative standard deviation of 73% (best 0.570, worst 0.038 across the ten runs), consistent with the 56% of the published five-run results [2]. Within a single run the training trajectory is equally erratic: in a 50-epoch exploratory run ADP stayed near zero for fifteen epochs, spiked to 0.899 at epoch 31, and collapsed again by epoch 35. A single-run comparison against a published number is therefore uninterpretable for this system.

Part of that variance arises after training ends. Re-evaluating one fixed trained checkpoint under ten evaluation seeds left Velox and ORTHRUS bit-for-bit identical, but moved NodLink’s ADP between 0.278 and 1.000 (mean 0.790, s.d. 0.212), because its VAE samples at inference time. Its precision (0.00096–0.00101) and recall (0.824–0.882) barely moved under the same reseeds. Two conclusions follow: handing over a trained NodLink model does not reproduce its ADP, and ADP is far more seed-sensitive than precision or recall.

Which threshold rule is applied matters as much as the model. Evaluating the same ten runs’ scores under three rules the framework offers moves mean precision by two orders of magnitude and mean recall from 0.985 to 0.018 (Table II): the configured NodLink rule floods (51,604 false positives on average), the leakage-prone MAGIC rule floods further, and a maximum-validation-loss rule all but silences the system. Reported detection performance is a property of the threshold rule as much as of the learned model.

The cause of the gap is identifiable. Abrar et al. [1] documented two inconsistencies between the NodLink paper and its released code: true positives counted from all cached subgraphs rather than only alert-triggering ones, which inflates reported precision, and a cache eviction policy in place of the paper’s time-decay rule. Inspection of the framework’s

implementation establishes a third fact: PIDSMaker removed the mechanism those defects belong to, retaining the VAE anomaly-scoring core with a threshold applied directly to node scores and no subgraph cache, Steiner-tree construction, or investigation stage. Three structurally different implementations therefore share the name “NodLink”: the paper’s system, whose reported evaluation contains the two documented defects; the released code, which Abrar et al. could not run on DARPA TC at all; and the framework’s reduced core, which produces the results above. The framework’s numbers measure the scoring core, not the published system, and no available implementation both runs on the benchmark and matches the paper’s description.

RQ1’s answer is that the published result does not reproduce through the common framework. Two independent reproductions agree two orders of magnitude below the paper’s precision, and the claimed operating point is unreachable at any threshold. What a reproduction obtains depends on four choices: the threshold rule, the epoch reported, the training seed, and, for NodLink specifically, the evaluation seed. Any reported number for this system is interpretable only alongside those four choices.

2) *RQ2: Four Systems on Operational Data:* On the full-density day (D8), no system offers a usable operating point as configured (Table III). NodLink catches every attack event (recall 1.0, 54/54 incidents) by raising 215,467 alarms, a false-discovery rate of 98.6%: an analyst clears 70 false alarms for every real one. ORTHRUS raises only 1,204 alarms at the best precision of the four (0.375) but reaches 17 of 54 incidents; 37 attacks produce no alarm at all. Velox raises 26,422 alarms to touch 8 incidents, and R-CAID flags a single benign node. All four screened at best ADP 0.70–0.90 on the DARPA benchmark (Table I); none of that performance survives contact with operational data.

Per-incident analysis shows which system is “best” depends entirely on the alarm budget an operation can afford. At 272 alarms, oracle-epoch Velox touches 41 of 54 incidents and everything else almost nothing; at 1,204 alarms NodLink leads the final-epoch systems with 29; at 26,422 NodLink saturates all 54 and even R-CAID’s ranking reaches 30. Velox, ORTHRUS, and R-CAID are also redundant rather than complementary: ORTHRUS detects every incident final-epoch Velox does, and at their oracle epochs adding ORTHRUS to Velox gains exactly one incident (41 to 42). Setting aside NodLink, whose flood catches everything by volume, twelve incidents are caught by no other system under any configuration, and they share a profile: mid-size campaigns too distributed to look compact yet too small to stand out, plus late small incidents sitting just under ORTHRUS’s threshold. Deploying several of these systems together would add almost nothing over deploying the best one.

The attacks themselves explain much of this. Clustering the 54 incidents on their measurable characteristics (events, hosts, duration, timing; k -means, $k=2$, silhouette 0.54) recovers exactly the two attack shapes in the data: 24 compact (two-to-five hosts) incidents and 30 wide fan-out campaigns.

Group membership predicts Velox’s detections (31% of wide campaigns versus 0% of compact ones) and predicts where R-CAID ranks attacks (wide campaigns near the top of its scoring, compact ones buried, though its threshold converts none of it into detections). For ORTHRUS the clusters predict nothing (31% versus 32%). Which attacks are detectable is largely a property of the attack, not the system. The baseline follows the same split: at the matched 26,422-alarm budget it covers 29 of 30 wide campaigns but only 10 of 24 compact incidents, whose few events pass between two hosts that already communicate routinely and so give the rarity and burst features nothing to fire on.

Separating each system’s ranking from its threshold rule locates the failures. Giving every system a perfectly chosen threshold in hindsight (the rank of its last attack node, the minimum cost of recall 1.0, Table V) shows NodLink’s ranking holds every attack inside its top 0.7% of nodes at D8: 14,395 alarms at precision 0.211, against 215,467 at 0.014 from its configured rule. Fifteen times the necessary alarms at D8, and 23 times at D4, come from the rule, not the ranking; the density degradation attributed to NodLink largely belongs to its threshold rule. The other three fail differently: Velox’s rule snaps between adjacent epochs (flags collapsing from $\sim 13,800$ to 64 at 2M while the underlying ranking barely moves; Table IV) over a ranking that needs 105,329 alarms for full recall at D8; ORTHRUS’s rule cannot reach its buried ranking, which holds full recall only beyond roughly a third of the graph; and R-CAID’s single-flag rule sits on a near-random ranking no rule could save. Across all four systems the configured threshold rule fails, each differently; for NodLink, and for Velox at its stronger epochs, the rule is the only component standing between the reported result and a usable one, while ORTHRUS’s and R-CAID’s rules sit on rankings too weak for any rule to rescue.

For two systems, whether a given attack is detected is close to arbitrary. ORTHRUS assigns its 2.08M nodes only 183 distinct scores, with one tied block holding 62% of all nodes, and Velox behaves similarly. Two near-identical incidents (73 and 74 events, the same two hosts, nine seconds each, 24 seconds apart) landed at the 99.98th percentile (detected) and the 64.4th (missed), and nothing about the attacks predicts which.

Density changes behaviour, not just numbers. NodLink degrades quantitatively: its quantile rule’s flag count grows with data volume (23,603 at 2M, 215,467 at D8, 590,336 at D4), so recall holds at 1.0 while the false-discovery rate climbs from 87.0% to 98.6% to 99.5%. ORTHRUS changes behaviour outright, inverting from near-full recall at 2M (0.937, 51/54 incidents) to starvation at D8 (0.148, 17/54) to silence at D4 (zero alarms). A benchmark number measured at one density does not predict the same system two densities away.

The conventional baseline (the five-feature Isolation Forest of Section IV-B) bounds these results from below. Its own decision rule is unusable, flagging 5.8M events, nearly half the day. Even with a perfectly chosen threshold, full recall costs 5% of the day’s 12M events at precision 0.005, so the

TABLE III

NODE-LEVEL DETECTION RESULTS AT THE FINAL TRAINING EPOCH WITH EACH SYSTEM’S CONFIGURED THRESHOLD RULE, ACROSS DATASET SCALES.

Dataset	System	Alarms	TP	Precision [95% CI]	Recall [95% CI]	F1	ADP	Inc.
2M	NodLink	23,603	3,062	0.130 [0.127, 0.133]	1.000 [1.000, 1.000]	0.230	0.198 ^a	54 ^a
	ORTHRUS	13,253	2,868	0.216 [0.211, 0.220]	0.937 [0.929, 0.946]	0.352	0.243	51
	Velox	77	0	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	0.000	0.177	0
	R-CAID	5	0	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	0.000	0.137	0
D8	NodLink	215,467	3,040	0.014 [0.014, 0.015]	1.000 [1.000, 1.000]	0.028	0.258	54
	ORTHRUS	1,204	451	0.375 [0.351, 0.396]	0.148 [0.136, 0.158]	0.213	0.178	17
	Velox	26,422	218	0.008 [0.007, 0.009]	0.072 [0.062, 0.079]	0.015	0.118	8
	R-CAID	1	0	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	0.000	0.004	0
D4	NodLink	590,336	3,040	0.005 [0.005, 0.005]	1.000 [1.000, 1.000]	0.010	0.136	54
	ORTHRUS	0	0	—	—	—	0.095	0

^a The NodLink 2M artifact predates the 54-attack ground-truth split and carries a single-group mapping; its ADP is computed over that mapping and its incident coverage (54/54) follows from recall 1.0. Velox and R-CAID were not run at D4 (resource decision; Section IV-B).

TABLE IV

THREE VIEWS OF THE SAME RUNS: ACROSS-EPOCH MEAN $\pm$ S.D. (SIX SAVED EPOCHS), THE FINAL TRAINING EPOCH (THE HEADLINE RULE), AND THE MAXIMUM TEST-SET-ADP EPOCH, LABELLED ORACLE BECAUSE NO DEPLOYMENT COULD SELECT IT.

Dataset	System	Mean $\pm$ s.d. (6 epochs)	Final		Oracle	
			ADP	Inc.	ADP (ep.)	Inc.
2M	NodLink	0.175 $\pm$ 0.029	0.198	— ^a	0.198 (9)	— ^a
2M	ORTHRUS	0.266 $\pm$ 0.026	0.243	51	0.314 (5)	51
2M	Velox	0.229 $\pm$ 0.047	0.177	0	0.284 (1)	54
2M	R-CAID	0.064 $\pm$ 0.060	0.137	0	0.137 (9)	0
D8	NodLink	0.246 $\pm$ 0.020	0.258	54	0.275 (0)	54
D8	ORTHRUS	0.132 $\pm$ 0.058	0.178	17	0.227 (1)	16
D8	Velox	0.153 $\pm$ 0.191	0.118	8	0.502 (7)	41
D8	R-CAID	0.002 $\pm$ 0.001	0.004	0	0.004 (9)	0
D4	NodLink	0.126 $\pm$ 0.019	0.136	54	0.156 (0)	54
D4	ORTHRUS	0.104 $\pm$ 0.040	0.095	0	0.159 (0)	4

^a Single-group ground-truth mapping (pre-split artifact); per-incident coverage not comparable. At the oracle epoch Velox@2M reaches recall 1.000 at precision 0.221 (13,848 alarms) and Velox@D8 precision 0.397 at 272 alarms. Both collapse by the final epoch.

TABLE V

THE MINIMUM COST OF RECALL 1.0 AT D8 (FINAL EPOCH): EACH SYSTEM’S RANKING GIVEN A PERFECTLY CHOSEN THRESHOLD IN HINDSIGHT (THE RANK OF ITS LAST ATTACK NODE), AGAINST ITS CONFIGURED RULE.

System	Hindsight (recall 1.0)		Configured rule	
	Alarms	Precision	Alarms	Recall
NodLink	14,395	0.211	215,467	1.000
Velox	105,329	0.029	26,422	0.072
ORTHRUS	741,493	0.004	1,204	0.148
R-CAID	2,046,352	0.001	1	0.000

ORTHRUS’s and R-CAID’s hindsight costs are $\sim 36\%$ and $>99.9\%$ of all scored nodes: full recall is unreachable at any usable cost regardless of rule.

attacks are not trivially catchable upstream: the dataset is a fair test. At matched alarm budgets, however, the baseline covers 39 of 54 incidents with 26,422 flags, more than ORTHRUS (21), R-CAID (30), or final-epoch Velox (8) achieve at the same budget, with the events-versus-nodes mismatch and the baseline’s 8x larger benign search space stated in the

Method. Only NodLink (54/54) clearly outperforms a five-feature detector with no graph and no deep learning on this data.

RQ2’s answer is that benchmark results do not transfer to this operational data. What determines behaviour there is not model architecture but three things the papers treat as fixed: the threshold rule, which fails in all four systems; the data’s density; and the match between each system’s inductive bet and the attacks’ shape. One caveat bounds all of it: the day’s attacks are short scripted bursts, not the campaigns PIDS are designed for (Section V-B), so these results characterise behaviour on operational telemetry rather than delivering a final verdict on the systems’ value against the threats they target.

B. Discussion

The central conclusion is that the configured threshold rule fails in every system, and each fails differently: NodLink’s quantile rule over-flags a strong ranking by a factor of fifteen at D8, Velox’s rule collapses between adjacent epochs while its ranking barely moves (Table IV), ORTHRUS’s rule cannot reach a ranking in which full recall sits a third of the graph deep, and R-CAID’s single-flag rule sits on a near-random ranking that no rule could save (Table V). The practical implication for system designers is that a published operating point bundles two components of different quality: the model’s ranking and the rule that cuts it. Reporting the ranking’s quality and its cost of recall alongside the configured threshold would let a deployment keep the ranking and substitute its own rule; for NodLink that substitution alone raises precision at full recall from 0.014 to 0.211. The implication for evaluation is that scale-sensitive rules, such as quantile cuts, should be anchored to score space before any cross-dataset comparison.

The tied-score structure undermines the field’s ranking metrics on ORTHRUS and Velox. ADP and average precision assume the ranking is meaningful all the way down, but within a tied group the order is arbitrary; at D8, ORTHRUS’s largest tie block holds 62% of all nodes, so the metrics partly measure a shuffle. The same model, day, and scores yield ADP between 0.998 and 0.243 depending only on evaluation granularity and

ground-truth grouping: the published-style edge-level single-attack mapping against the node-level 54-attack mapping (Table VII, Appendix B). A second mechanism affects NodLink: evaluation seeds reorder the top of its ranking while barely changing which nodes cross the threshold, so ADP swings under reseeding while precision and recall hold still. Epoch selection adds a third: the framework’s default of reporting the best test-set epoch is a form of leakage, and the final-epoch rule with a labelled oracle bound and an across-epoch mean $\pm$ s.d. (Table IV) is proposed as the honest alternative. None of this makes ADP useless; it makes an ADP number uninterpretable without its tie structure, grouping, granularity, and epoch rule.

The recall-versus-precision tension resolves into a risk/benefit judgment rather than a winner. Under a recall-first reading only NodLink’s flood survives, at a false-discovery rate of 98.6% as configured and 79% even perfectly tuned. Under a precision-first reading only ORTHRUS’s configured point and oracle-epoch Velox’s 272-alarm point survive, and both miss most incidents. On this data every system either accepts most of the risk or delivers very little benefit, and alarms beyond triage capacity are never analysed, so NodLink’s full recall as configured holds on paper rather than in a triage queue.

Why particular attacks are missed is largely attributable to the attacks: the two attack shapes predict Velox’s detections and R-CAID’s rankings, and the twelve incidents caught by nothing but NodLink’s flood sit between the shapes, matching neither bet. At the featurisation layer, all 202 distinct tokens appearing in attack events are also common in benign events, so text embeddings offer the models little purchase. Not every pattern is explained: ORTHRUS and Velox detect only the earliest of the day’s fan-out campaigns, a strict prefix in time with a shared cutoff, while identically sized campaigns after the cutoff are missed. Graph-construction window membership was checked and ruled out, leaving sequential-inference behaviour, in which repeated near-identical campaigns stop registering as anomalous, as an untested hypothesis.

Training data quality proved to be about content, not labels. Adding former attack days to training (their malicious events removed) destabilised ORTHRUS, and the collapse persisted when the enriched set was sampled back to the baseline’s volume, isolating the cause to the content of ex-attack days rather than data quantity. For operational deployments this is an awkward requirement: cleaning the attacks out of a day is not enough; the day must never have contained them, and a deployment under regular attack may have no such history.

The dataset bounds what can be concluded. Every one of the 54 attacks is a scripted burst lasting under ten seconds, carrying a single lifecycle stage and one event type. The day holds exactly two attack shapes, and nothing resembles the multi-stage, low-and-slow campaign PIDS are designed to catch [3], [9]. The results therefore support conclusions about how these systems behave on operational telemetry, but not a final verdict on their value against the threats they actually target. A PIDS’s distinctive value at the last line of detection

is catching attacks visible only in context; a day whose attacks a per-event detector largely reaches cannot measure that value, and the compact incidents, which the baseline buries and ORTHRUS partially recovers, are the one place it shows.

Four threats to validity bound these conclusions. Ground truth is operational, recording the correlation engine’s own verdicts rather than analyst-confirmed labels, so systematic engine blind spots would be invisible. Each result cell is a single run, and bootstrap confidence intervals cover sampling variance but not training variance. ORTHRUS is demonstrably nondeterministic run to run, and the RQ1 work shows single runs can mislead, so cross-system conclusions rest on effect sizes far larger than the observed instability rather than on repeated runs. The NodLink 2M artifact predates the 54-attack ground-truth split, so its per-incident values are not comparable at that scale. And the baseline comparison, where included, matches alarm budgets across different flagged populations (events against nodes), so only its large differences are read.

The questions are answered to a bounded extent: RQ1 in full for one system on one benchmark, with no claim of generality to the seven systems only screened; RQ2 for one organisation’s test day at three densities, with whether the same properties dominate on other organisations and longer campaigns left to future work.

VI. CONCLUSION AND FUTURE WORK

This project asked whether published PIDS results reproduce, and whether they transfer to operational data. The answer to both is no. NodLink’s published CADETS_E3 result is unreachable at any threshold on the common framework’s scores, two independent reproductions agree, and the causes trace to divergences between the paper, its released code, and the framework’s reduced reimplementations. On a real operational day, four systems that screened at best ADP 0.70–0.90 on the benchmark produced no usable operating point as configured, and their behaviour was determined less by architecture than by threshold rules, data density, and the match between each system’s inductive bet and the attacks’ shape. The constructive core is the separation of rankings from rules: for the systems with strong rankings, what looks like model failure is threshold-rule failure; given a perfectly chosen threshold, NodLink holds every attack in its top 0.7% of nodes, fifteen times fewer alarms than its own rule raises.

Future work follows directly. The second organisation in the export (ORG-0005) would test the training-content rule in an environment with no clean history; the unsampled test day (12M events) awaits hardware beyond the cluster’s memory ceilings; mappings A and C remain to be evaluated empirically against mapping B; the per-epoch checkpointing is a candidate upstream contribution to PIDSMaker; and comparing the systems’ flagged subgraphs against Precinct’s own incident case graphs would extend the evaluation from detection to investigation, the stage the framework currently omits.

REFERENCES

- [1] T. Abrar, A. Shamil, M. J. Iqbal, A. Ahmed *et al.*, “On the reproducibility of provenance-based intrusion detection that uses deep learning,”

in *ACM Conference on Reproducibility and Replicability (ACM REP)*, 2025.

- [2] T. Bilot, B. Jiang, Z. Li, N. El Madhoun, K. Al Agha, A. Zouaoui, and T. Pasquier, “Sometimes simpler is better: A comprehensive analysis of state-of-the-art provenance-based intrusion detection systems,” in *USENIX Security Symposium*. USENIX Association, 2025.
- [3] Y. Xiao, L. Jiang, S. Nie, D. Li, S. Wu, K. Xu, and Q. Li, “How far should we need to go: Evaluate provenance-based intrusion detection systems in industrial scenarios,” *arXiv preprint arXiv:2603.22982*, 2026.
- [4] T. Bilot, B. Jiang, and T. Pasquier, “PIDSMaker: Building and evaluating provenance-based intrusion detection systems,” *arXiv preprint arXiv:2601.22983*, 2026.
- [5] S. Li, F. Dong, X. Xiao, H. Wang, F. Shao, J. Chen, Y. Guo, X. Chen, and D. Li, “NodLink: An online system for fine-grained APT attack detection and investigation,” in *Network and Distributed System Security Symposium (NDSS)*, 2024.
- [6] WitFoo, Inc., “Precinct6 cybersecurity dataset,” <https://huggingface.co/datasets/witfoo/precinct6-cybersecurity>, 2026, revisions pinned: 2M cc7123d1, 100M f63ffdbc.
- [7] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in *IEEE Symposium on Security and Privacy (S&P)*, 2010.
- [8] W. U. Hassan, A. Bates, and D. Marino, “Tactical provenance analysis for endpoint detection and response systems,” in *IEEE Symposium on Security and Privacy (S&P)*, 2020.
- [9] F. Dong, S. Li, P. Jiang, D. Li, H. Wang, L. Huang, X. Xiao, J. Chen, X. Luo, Y. Guo, and X. Chen, “Are we there yet? An industrial viewpoint on provenance-based endpoint detection and response tools,” in *ACM Conference on Computer and Communications Security (CCS)*, 2023.
- [10] Y. Meng, S. Li, J. Gui, P. Jiang, and D. Li, “KnowHow: Automatically applying high-level CTI knowledge for interpretable and accurate provenance analysis,” in *Network and Distributed System Security Symposium (NDSS)*, 2026.
- [11] M. A. Inam, Y. Chen, A. Goyal, J. Liu, J. Mink, N. Michael, S. Gaur, A. Bates, and W. U. Hassan, “SoK: History is a vast early warning system: Auditing the provenance of system intrusions,” in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2023.
- [12] 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 *USENIX Security Symposium*. USENIX Association, 2025.
- [13] A. Goyal, G. Wang, and A. Bates, “R-CAID: Embedding root cause analysis within provenance-based intrusion detection,” in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2024.
- [14] A. Goyal, X. Han, G. Wang, and A. Bates, “Sometimes, you aren’t what you do: Mimicry attacks against provenance graph host intrusion detection systems,” in *Network and Distributed System Security Symposium (NDSS)*, 2023.
- [15] K. Mukherjee, J. Wiedemeier, T. Wang, J. Wei, F. Chen, M. Kim, M. Kantarcioglu, and K. Jee, “Evading provenance-based ML detectors with adversarial system actions,” in *USENIX Security Symposium*, 2023.
- [16] Y. Huang, S. Li, Y. Guo, X. Chen, D. Li, and W. U. Hassan, “No data? no problem: Synthesizing security graphs for better intrusion detection,” *arXiv preprint arXiv:2506.06226*, 2025.
- [17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *IEEE International Conference on Data Mining (ICDM)*, 2008.
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

APPENDIX A

PROJECT MANAGEMENT EVIDENCE

Time spent on the project was tracked in Clockify. Table VI records the hours spent per milestone against the plan report’s planned hours, and Figure 1 plots cumulative hours against the plan’s expectation.

A. Tasks as Defined in the Plan Report

Table VI restates the plan report’s task decomposition, with planned hours, actual hours, and the outcome. Deviations beyond the $\pm 30\%$ threshold are marked † and explained below.

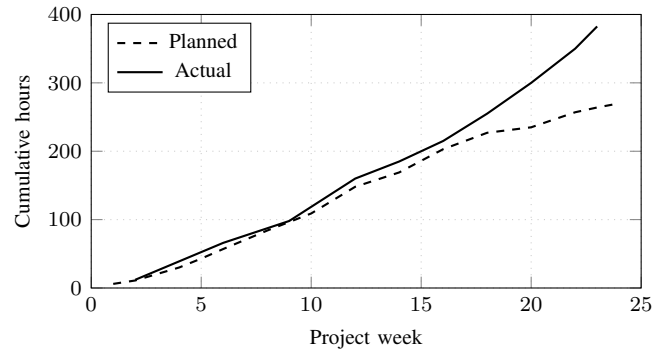


Fig. 1. Cumulative project hours against the plan report’s expectation. The curves track together until roughly week 15; the divergence thereafter is the M6 expansion, when the WitFoo generalisation study became the project’s main thrust.

B. Significant Deviations

Five deviations cross the significance threshold; each traces to a documented supervisor decision or an explicit scope trade.

M3 — RQ1 went deeper, not wider (+58%). The plan budgeted 38 hours to evaluate NodLink on all three DARPA TC E3 datasets. By supervisor agreement the evaluation was confined to CADETS_E3 and the saved hours redirected into depth: the threshold analysis across all saved score files, the codebase investigation that located the architectural divergence between the framework’s NodLink and the paper’s, and the checkpoint-reseed demonstration separating inference-time from training-time variance. The breadth-for-depth trade produced RQ1’s main findings and still overran, reflecting the learning curve the plan anticipated.

M5 — the comparative benchmark changed shape. The planned NodLink-versus-KAIROS comparison was superseded in May by supervisor decision, after the published five-run comparison of Bilot et al. covered the same ground: KAIROS was dropped and the milestone became an eight-system screening on CADETS_E3 (one 50-epoch exploratory run per system) that selected the four systems studied in full. Hours were close to plan; the deliverable is different from, and broader than, the one specified.

M6 — the generalisation study became the project (+180%). The plan treated real WitFoo data as an extension that might not materialise (risk R3: likelihood medium, impact low) and budgeted 48 hours for synthetic structural variants of DARPA TC. The opposite occurred: at the July supervisor meeting the project’s main thrust was redirected onto the WitFoo operational dataset, and synthetic variants were abandoned in favour of a natural experiment on real enterprise days. The milestone absorbed the translation pipeline and its three label mappings, per-incident ground truth, the dataset census and quality findings reported upstream to WitFoo, the full four-system campaign at three densities, and the per-incident analysis. Most of the project’s contribution now sits in this milestone, which also absorbed the time freed by M11.

M11 — the adversarial stretch goal was dropped. The interim checkpoint recorded the rationale: the systems proved

TABLE VI
THE PLAN REPORT’S TASK DECOMPOSITION (ITS TABLE III) AGAINST ACTUAL HOURS AND OUTCOMES. ACTUAL HOURS ARE TRACKED IN CLOCKIFY AT MILESTONE LEVEL. † MARKS DEVIATIONS BEYOND $\pm 30\%$ OR A CHANGED DELIVERABLE.

MS	Task (as defined in the plan report)	Plan	Actual	Outcome
M1	Read and summarise NodLink, KnowHow, and Abrar et al.	22	70	Delivered
M1	Read and summarise PIDSMaker and KAIROS papers	12		
M1	Review NodLink code-base	15		
M1	PIDSMaker environment setup; DARPA TC E3 end-to-end	20		
M2	Plan report: draft, appendix, review, submit	15	19.5 [†]	Delivered (W5)
M3	Run NodLink via PIDSMaker on DARPA TC E3; comparison table; discrepancies classified	38	60 [†]	Delivered, deeper scope
M4	Interim checkpoint: prepare and deliver presentation	15	15	Delivered (W12, passed)
M5	Run KAIROS via PIDSMaker; head-to-head comparison	42	40 [†]	Deliverable changed
M6	RQ3 structural variation: design experiments; generate variants; sensitivity analysis	48	135 [†]	Deliverable changed; became main thrust
M7	Final report: synthesise all findings; complete IEEE report	30	40 [†]	Delivered
M8	Showcase abstract for FYP booklet	3	3	Delivered (W21)
M9	CSSE FYP Showcase: prepare and deliver presentation	5	—	Pending (W24)
M10	Final demo: prepare live demonstration and Q&A	12	—	Pending (exam period)
M11	[Stretch] Mimic-Prov evasion runs at four insertion ratios	20	0 [†]	Dropped by supervisor decision
Plan total (excl. / incl. stretch)		277 / 297		
Total excl. pending M9/M10 (incl. stretch)		280	382.5	

too fragile under standard conditions for evasion experiments to add meaning. The decision was made with supervisors and its 20 planned hours moved to M6.

M2 and M7 — reporting overheads (+30% and +33%). The plan report’s appendix and figures took longer than budgeted, and the final report’s writing load grew with the volume of results it must carry; both overruns are absorbed within the overall hours.

C. Unplanned Deliverables

Four artefacts were produced that the plan did not anticipate: a checkpointing and reseeding capability on the PIDSMaker fork (per-epoch model saves, reframed by supervisors as a reproducibility hand-off mechanism); the WitFoo-to-PIDSMaker

translation pipeline with streaming constant-memory operation and deterministic benign sampling; the 54-incident characterisation joined to every system’s detections; and a conventional-ML baseline (Isolation Forest) for calibrating the PIDS results.

D. Supervision and Collaboration Record

Supervision ran as a weekly meeting with minutes kept in the shared project drive; course-directed artefacts (plan report, interim checkpoint) were delivered on their fixed weeks. The collaboration with WitFoo ran by correspondence throughout, including the dataset-defect reports that led to a corrected dataset release.

APPENDIX B METRIC SENSITIVITY DETAIL

Table VII records the full set of evaluation choices behind the ADP sensitivity discussed in Section V-B: one model, one test day, four scores.

TABLE VII
ONE MODEL, ONE TEST DAY, FOUR SCORES: ORTHRUS AT 2M, FINAL EPOCH, UNDER DIFFERENT EVALUATION CHOICES.

Evaluation choice	Metric	Score
Edge-level, single-attack mapping	ADP	0.998
Node-level, 54-attack mapping	ADP	0.243
Node-level	Entity AP	0.233
Edge-level	Entity AP	0.221
Node-level, configured threshold	Precision / Recall	0.216 / 0.937
Edge-level, configured threshold	Precision / Recall	0.251 / 0.411