

Does It Generalize? A Cross-Dataset Study of Graph-Based Insider Threat Detection Beyond CERT

QASIM Mohamed Muhanna Alriyami, Mohd Murtadha Bin Mohamad
Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia

Abstract—Graph-based insider threat detection methods increasingly report strong performance by modelling relationships among users, hosts, resources and security events. However, many studies still evaluate within a single benchmark family—usually CERT—or report separate within-dataset results, so it is unclear whether the learned representations capture transferable insider behaviour or dataset-specific artefacts. This study presents, to the best of our knowledge, the first systematic zero-shot cross-dataset benchmark for insider threat detection across CERT, SPEDIA and LANL. In this setting, no labelled or unlabelled target observations are used during training. Events are harmonised to a common user-day prediction unit with label-independent event sampling, and in-distribution performance is controlled with both chronological and user-disjoint splits. Within-domain performance of a Random Forest reference model was heterogeneous (user-disjoint PR-AUC from 0.004 on LANL to 0.99 on SPEDIA). Across evaluable targets, its mean in-distribution PR-AUC was 0.542, compared with a mean off-diagonal zero-shot PR-AUC of 0.133, yielding a generalization gap of 0.409. A memory-based temporal graph neural network without explicit time encoding was compared with this reference on two transfers into SPEDIA using paired seed-level bootstrap intervals. Representation effects were source-dependent: the temporal graph model outperformed the Random Forest for CERT $r4.2 \rightarrow SPEDIA$ ($\Delta PR = +0.123$, 95% CI $[+0.075, +0.169]$), whereas the Random Forest outperformed the graph model for CERT $r5.2 \rightarrow SPEDIA$ ($\Delta PR = -0.031$, 95% CI $[-0.061, -0.012]$). The results therefore do not support a general advantage of graph representations. They show that strong single-dataset performance is insufficient evidence of cross-dataset generalization.

Keywords—Insider threat detection; cross-dataset generalization; zero-shot evaluation; graph neural networks; temporal graphs; self-supervised learning; CERT; SPEDIA; LANL

I. INTRODUCTION

Insider threats remain difficult to detect because malicious, negligent, or compromised users may operate through legitimate accounts, authorised devices, and routine organisational workflows. Unlike many external attacks, insider activity does not necessarily produce an immediate perimeter violation. Its significance may instead emerge through a sequence of individually plausible actions involving authentication, host access, files, removable devices, communications, and other enterprise resources. The difficulty of obtaining suitable insider-threat data, together with confidentiality and privacy constraints on organisational logs, further complicates the development and evaluation of reliable detectors [1].

Machine-learning and deep-learning methods have improved the ability to model these behavioural signals. Recent graph-based approaches are especially promising because enterprise activity can be represented as interactions among

users, hosts, resources, and timestamped events. Graph neural networks can propagate relational context between connected entities, while temporal graph models can retain changes in behaviour across ordered interactions. ATHITD uses a temporal heterogeneous graph to represent insider activity, whereas Log2Graph transforms security logs into graph structures for graph-convolutional detection [2], [3]. These developments extend insider threat detection beyond isolated records and fixed aggregated feature vectors.

However, methodological progress has not been matched by equally rigorous evidence of cross-dataset transfer. The synthetic CERT corpus remains the dominant experimental benchmark, and several recent sequence, relational, and graph-based studies continue to evaluate their methods using only CERT-derived data or, like ATHITD and Log2Graph, CERT and LANL without cross-dataset testing [2]–[6]. Reviews also document frequent use of CERT in insider-threat research [7]. CERT provides controlled scenarios, multiple enterprise event sources, and answer-key labels, making it valuable for reproducible model development [1]. Nevertheless, a detector trained and tested within one CERT release may learn scenario-specific frequencies, user-role patterns, event-generation rules, or other corpus regularities that are absent from independently constructed datasets. Strong within-dataset performance therefore does not establish that the learned representation will remain effective in a different organisational or logging environment.

Network intrusion detection has addressed this concern more directly. Cross-dataset studies have demonstrated that models performing strongly on a source benchmark may deteriorate when evaluated on independently collected network traffic [8], [9]. Together, these studies motivate source–target evaluation matrices for comparing within-dataset and cross-dataset performance [8], [9], while Sarhan et al. examine a common feature set for cross-environment generalisability [10]. (A source–target matrix scores every train-on-one, test-on-another dataset pair; a harmonised representation uses one shared feature set for all datasets.) Zero-shot transfer, which applies a trained model to a new dataset without retraining or tuning on it, has also been studied under protocol and dataset shift [11]. Cross-dataset evaluation is consequently treated as a distinct benchmark rather than as an extension of a random train–test split. Insider threat detection has not yet adopted an equivalent evaluation paradigm. To the best of our knowledge, prior graph-based insider-threat studies have not evaluated transfer from CERT to an independently constructed target under a strict zero-shot protocol, and systematic source-to-target benchmarking across independently

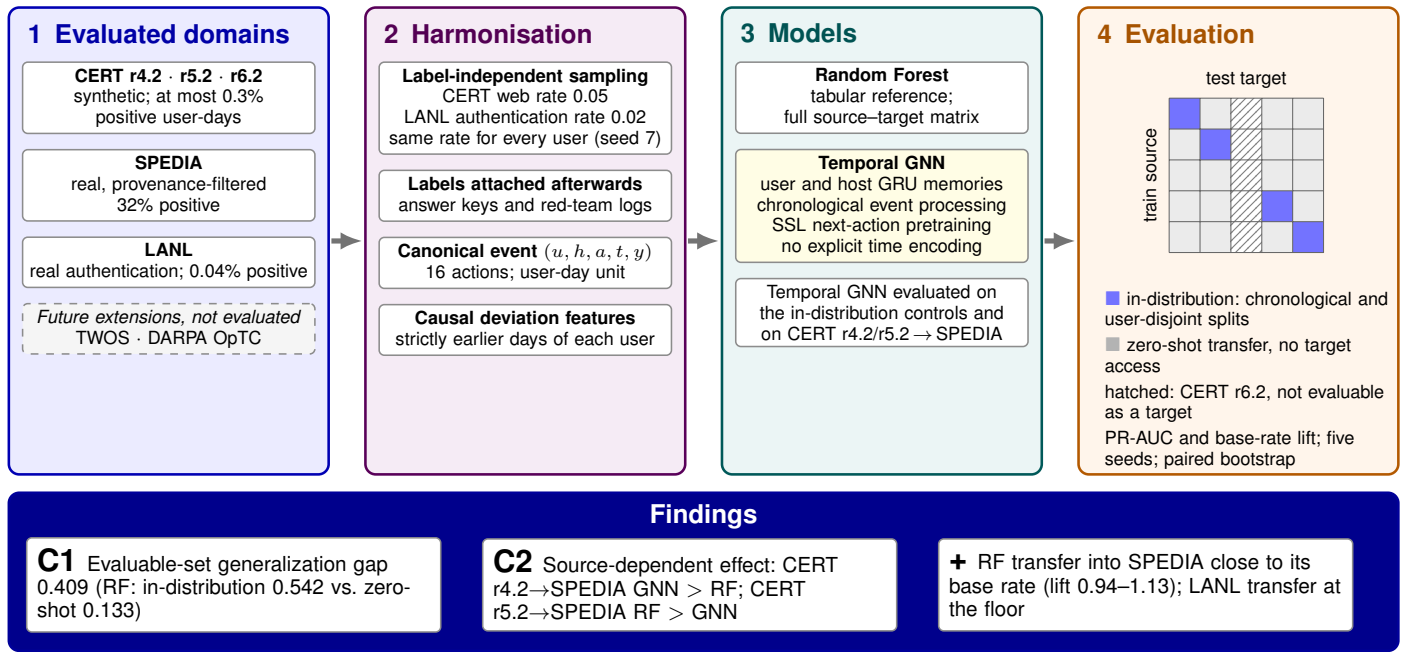


Fig. 1. Overview of the cross-dataset benchmark. Three evaluated domain families are harmonised to a canonical event schema and a user-day prediction unit with label-independent event sampling. A Random Forest reference model is evaluated on the full source-by-target matrix, and a memory-based temporal graph neural network without explicit time encoding is evaluated on the in-distribution controls and on two transfers into SPEDIA. The findings panel summarises the evaluable-set generalization gap and the source-dependent representation comparison.

constructed insider-threat datasets remains limited. It therefore remains unclear whether recent graph models learn transferable insider-threat behaviour or dataset-specific decision rules. This uncertainty is particularly important because CERT, SPEDIA, and LANL differ in data origin, event coverage, insider type, labelling procedure, population, and class prevalence [1], [12], [13].

This study addresses the gap with a *zero-shot cross-dataset generalization* benchmark: a model is trained on a source dataset and evaluated on an independently defined target dataset without access to labelled or unlabelled target observations. Target-assisted domain adaptation is outside the scope of the reported experiments, so no result in this study uses target data during training. The principal contributions are as follows:

- A *strict zero-shot cross-dataset benchmark*. We present, to the best of our knowledge, the first systematic zero-shot cross-dataset benchmark for insider threat detection across CERT, SPEDIA and LANL. It combines explicit target isolation, label-independent event sampling and harmonisation, chronological and user-disjoint in-distribution controls, base-rate lift reporting and an evaluable-set generalization gap. For a Random Forest reference model, mean PR-AUC fell from 0.542 in distribution to 0.133 under zero-shot transfer ($\Delta = 0.409$).
- A *source-dependent comparison of tabular and temporal-graph representations*. Using paired seed-level bootstrap intervals, we compare the reference model with a memory-based temporal graph neural network on two transfers into SPEDIA. The graph model was better for CERT r4.2 as the source, and the

tabular model was better for CERT r5.2. Representation advantages therefore did not transfer consistently, even into the same target dataset.

The remainder of this study is organised as follows. Section II reviews insider threat detection, graph-based methods, cross-dataset network intrusion detection, and the selected datasets. Section III defines the datasets, prediction unit, and zero-shot benchmark. Section IV describes the temporal graph construction and detector. Section V describes the experimental protocol, and Section VI reports the findings. Sections VII and VIII discuss threats to validity and conclude the study, respectively. Fig. 1 summarises the overall framework.

II. RELATED WORK

A. Machine Learning and Deep Learning for Insider Threat Detection

Insider threat detection has commonly been formulated as supervised classification, anomaly detection, or behavioural profiling over enterprise activity records. Conventional methods aggregate events into user-level, session-level, or user-day features and apply classifiers or outlier detectors to distinguish suspicious activity from normal behaviour. These methods are computationally practical and can provide useful operational summaries. However, aggregation may obscure the event ordering, timing, and interaction context through which gradual insider behaviour becomes observable.

Deep-learning studies attempt to preserve more of this structure. Recurrent neural networks, autoencoders, attention mechanisms, and transformer-based models can represent behavioural sequences and identify deviations from historical activity. Temporal feature aggregation with attention has been

used to combine activity-log evidence across time [5], while relational multi-sequence modelling has been proposed to capture dependencies among different behavioural sources [6]. LAN learns adaptive neighbourhood relationships for real-time insider threat detection, illustrating the increasing use of relational learning beyond fixed feature aggregation [4]. However, existing machine-learning and sequence-based studies provide limited evidence that their learned behavioural representations remain effective outside the dataset on which they were developed.

B. Graph-Based Insider Threat Detection

Graph-based insider threat detection represents enterprise activity through relationships among users, hosts, resources, and security events. Such representations can preserve interaction structure that is lost when each user-day is treated as an independent feature vector. Static graph convolutional networks aggregate neighbourhood information, while graph attention mechanisms learn the relative importance of connected entities or event types. Temporal graph models additionally retain the order and timing of interactions.

Log2Graph converts security logs into graph structures and applies graph convolution for insider threat detection [3]. Its formulation demonstrates the value of representing log relationships explicitly rather than relying only on flat event aggregates. ATHITD is one of the closest works to the present study because it combines attention with a temporal heterogeneous graph representation for insider threat detection [2]. A subsequent fusion model integrates sequential and graph neural components with an explainability mechanism for user and entity behaviour analytics [14].

ATHITD and the fusion model are particularly relevant architectural comparators, but neither measures cross-dataset transfer. ATHITD reports results on CERT and LANL and the fusion model on CERT, yet neither trains on one insider threat dataset and evaluates on another, independently constructed dataset. Their reported results therefore measure within-dataset discrimination rather than zero-shot cross-dataset transfer. Log2Graph is subject to the same general limitation: it demonstrates a graph-based detector within its development setting but does not establish whether the learned representation transfers across dataset boundaries [3].

The present study does not reimplement these architectures. Instead, it evaluates a memory-based temporal graph detector and a tabular reference model under the same user-day labels, aligned event vocabulary, source–target partitions, and access restrictions. Consequently, existing graph-based detectors have not been compared systematically under a zero-shot source-by-target protocol that separates strong single-dataset performance from transferable graph learning.

C. Cross-Dataset Generalization and Domain Adaptation in Network Intrusion Detection

Network intrusion detection provides the closest established precedent for the present evaluation design. Layeghy and Portmann examined the generalisability of machine-learning network intrusion detectors and showed that strong performance on a development dataset does not necessarily persist under cross-dataset evaluation [8]. Sarhan et al. investigated standardised feature sets intended to improve both

generalisability and explainability across network intrusion datasets [10]. Cantone et al. further analysed cross-dataset generalization and the effects of distributional differences between independently collected benchmarks [9].

These studies motivate the use of a source–target matrix rather than a single within-dataset score. A model is trained on one source and evaluated on multiple targets, allowing diagonal in-distribution performance to be separated from off-diagonal transfer. Such evaluation exposes dataset bias that may remain hidden when training and testing records originate from the same collection process.

Domain adaptation addresses a related but different problem. Instead of requiring transfer without target access, adaptation methods use information from the target domain to reduce the distribution mismatch; because the target distribution participates in training, such methods address unsupervised domain adaptation rather than zero-shot generalization. Oladejo and Ahmed apply cross-domain transfer learning to multi-protocol intrusion detection, but their reported experiments are zero-shot, with no target fine-tuning or domain adaptation [11]. The present study likewise reports zero-shot results only.

Although the distinction between source-only cross-dataset generalization and target-assisted unsupervised adaptation is established in network intrusion research, it has not been applied systematically to graph-based insider threat detection.

D. Insider Threat Datasets

The CERT insider-threat corpus was created to support reproducible research when labelled organisational insider incidents were difficult to obtain [1]. It provides synthetic users, hosts, organisational roles, multi-source activity records, and answer-key labels for traitor scenarios. Its controlled structure and accessibility explain its widespread use, but its synthetic generation process may also introduce regularities that are not present in independently collected environments.

TWOS was developed from a gamified competition containing malicious insider behaviour [15]. It includes labelled masquerader and traitor activity spans and therefore differs from CERT in collection setting, scenario construction, and available event fields. These differences make TWOS a natural candidate for extending the present benchmark. It is therefore identified as a target for future work and is not evaluated in this study.

SPEDIA provides security-event data with behavioural descriptions and Wazuh severity information [12]. The release contains both real and synthetic components, including records associated with CERT. A clean CERT-to-SPEDIA comparison therefore requires provenance-aware filtering so that CERT-derived records do not appear in the target evaluation. In this study, every row for which `Decoder_name == 'cert'` is excluded. The audited remainder contains 20,619 real rows, zero occurrences of `dtaa.com`, zero `cert` tokens, and zero CERT-style identifiers. The retained records are drawn only from auditd, PAM, syscheck, and JSON sources.

The LANL cybersecurity dataset contains authentication and other enterprise events collected from a real organisational environment [13]. Its labelled red-team authentication activity differs semantically from CERT traitor scenarios and is more

closely associated with masquerader or lateral-movement behaviour. LANL nevertheless provides a valuable independently collected domain for testing whether models transfer beyond synthetic insider scenarios.

These datasets cannot be made semantically identical through preprocessing. CERT, SPEDIA, and LANL differ in event types, malicious-activity definitions, temporal coverage, and labelling granularity. The present work therefore aligns them at a common user-day prediction level while retaining and reporting their original threat semantics.

What is still missing is a harmonised benchmark that evaluates insider threat detectors across CERT, SPEDIA, and LANL without concealing the substantive differences in their labels and collection processes.

III. DATASETS AND PROBLEM FORMULATION

A. Datasets

1) *CERT*: CERT provides synthetic enterprise activity together with answer-key labels for traitor scenarios [1]. Releases r4.2, r5.2, and r6.2 are used. Events are mapped to user-day intervals, and a user-day is labelled positive when it contains at least one answer-key malicious event. CERT is used as a controlled source domain whose in-distribution performance is measured under both chronological and user-disjoint splits. The term “overfitting” is used only when these controls support that interpretation; otherwise, an off-diagonal decline is described as failure to transfer.

2) *SPEDIA*: SPEDIA contains security records with descriptive and Wazuh severity information [12]. A user-day is labelled positive when at least one event has a `Description` of “Highly Suspicious” or “Midly Suspicious”, or a Wazuh level of at least 8. To prevent CERT-derived overlap, all rows satisfying `Decoder_name == 'cert'` are excluded. The audited real-only subset contains 20,619 rows, zero `dtaa.com` occurrences, zero `cert` tokens, and zero CERT-style identifiers. Its retained sources are limited to `auditd`, `PAM`, `syscheck`, and `JSON` records.

3) *LANL*: The LANL dataset contains authentication and related events from a real enterprise environment, together with labelled red-team activity [13]. Each authentication event is mapped to the corresponding user and host within the aligned action vocabulary. A user-day is positive when it contains at least one labelled red-team authentication event. Because these labels principally represent masquerader or lateral-movement activity, LANL is treated as a related but not identical insider-threat domain. Authentication records of machine accounts are removed.

4) *Label-independent event sampling*: The CERT web logs (14.5–90 GB per release) and the LANL authentication log (about 1.05 billion rows) are too large to process in full, so these two event sources are subsampled. Every CERT web event is retained independently with probability 0.05, and every eligible LANL authentication event is retained independently with probability 0.02, using a fixed pseudo-random seed (7). The retention decision depends only on this random draw; it does not use user identity, the ground-truth insider or red-team rosters, or any label. User-day labels are attached only after the sampled event universe has been fixed.

All other CERT event sources are retained in full, and SPEDIA is not subsampled. A labelled user-day can be observed only if at least one of its events is retained; for LANL, 158 of the 176 red-team user-days remain observable after sampling.

5) *Evaluable targets*: A domain is evaluated as a target, and its in-distribution score is reported, only when it contains at least 50 positive user-days. CERT r6.2 contains 44 positive user-days among 1,394,010 user-days and is therefore used as a training source only. The other four domains meet the criterion.

B. Common Prediction Unit and Graph Domain

Let the benchmark contain K domains,

$$\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_K\}, \quad (1)$$

where, each domain corresponds to one dataset and defines a distribution over timestamped security events. An event is represented as

$$e = (u_e, h_e, a_e, t_e), \quad (2)$$

where, u_e is the user, h_e is the host, a_e is an action from the fixed vocabulary $\mathcal{A}$, and t_e is the event timestamp. All datasets are converted to the same user-day prediction unit. For user u and day d , let $\mathcal{E}_{u,d}$ denote the set of events assigned to that interval. Its binary label is

$$y_{u,d} = \mathbb{I}[\exists e \in \mathcal{E}_{u,d} \text{ such that } \ell_e = 1], \quad (3)$$

where, ℓ_e is the dataset-specific malicious-event or malicious-span indicator. This definition preserves each dataset’s documented labelling rule while enforcing a common unit of analysis. The user-day is the unit of labelling, prediction, and evaluation. It is not the input resolution of the temporal model. The temporal graph detector consumes individual timestamped events in chronological order, so event ordering and within-day temporal structure enter its memory state; the aggregated user-day features are used alongside that memory. Daily labels nevertheless have limitations. They cannot measure detection latency within a day, event-level detection behaviour, or differences between sessions. Event-, session-, host-, or process-level formulations should therefore be evaluated where the datasets support them.

Each domain is represented as a heterogeneous temporal graph

$$\mathcal{G}_i = (\mathcal{V}_i^U, \mathcal{V}_i^H, \mathcal{E}_i), \quad (4)$$

where, $\mathcal{V}_i^U$ and $\mathcal{V}_i^H$ are user and host nodes, respectively, and $\mathcal{E}_i$ contains action-typed timestamped edges. Dataset-specific raw events are mapped to the fixed action vocabulary before model fitting. Continuous feature normalisation is fitted using the permitted training data only.

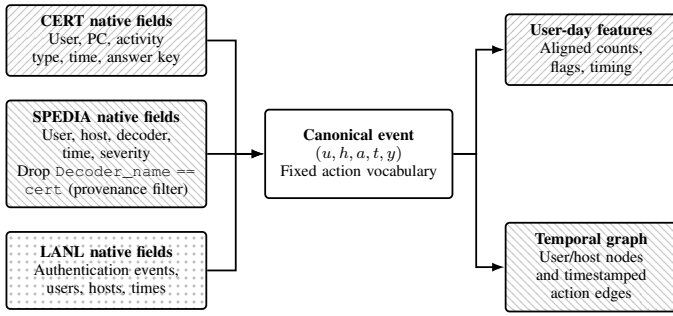


Fig. 2. Data harmonisation pipeline.

Fig. 2 illustrates the harmonisation pipeline.

C. Zero-Shot Cross-Dataset Generalization

The primary benchmark measures zero-shot cross-dataset generalization. For source domain i , a detector f_i^{ZS} is trained using only labelled source-domain training data:

$$f_i^{ZS} = \text{Train}(\mathcal{D}_i^L). \quad (5)$$

No labelled or unlabelled target events, target feature statistics, target validation results, or target-domain model-selection signals are available during training.

For source i and target j , the matrix entry is

$$M_{ij} = \text{PR-AUC}(f_i^{ZS}, \mathcal{D}_j^{\text{test}}). \quad (6)$$

The diagonal entries represent in-distribution evaluation, whereas $i \neq j$ denotes zero-shot transfer. In-distribution scores are estimated under both a chronological split and a user-disjoint split (Section V). The complete source-by-target matrix is computed for the Random Forest reference model, and the temporal graph detector is evaluated on the in-distribution controls and on two transfers into SPEDIA (Section VI).

Let S denote the set of source domains, $T \subseteq S$ the set of evaluable target domains, and

$$P = \{(i, j) : i \in S, j \in T, i \neq j\} \quad (7)$$

the set of evaluable transfer cells. The generalization gap is defined as

$$\Delta = \frac{1}{|T|} \sum_{j \in T} M_{jj} - \frac{1}{|P|} \sum_{(i,j) \in P} M_{ij}. \quad (8)$$

The first term is the uniformly weighted mean of the evaluable in-distribution cells, and the second is the uniformly weighted mean of the evaluable zero-shot cells. A larger positive value indicates a greater decline from in-distribution performance to zero-shot transfer. CERT r6.2 remains available as a training source but is excluded from T because its target evaluation does not meet the positive-count criterion; with five sources and four evaluable targets, $|P| = 16$. The in-distribution cells in Eq. (8) use the user-disjoint split, because

the chronological split leaves no LANL positives in the test period.

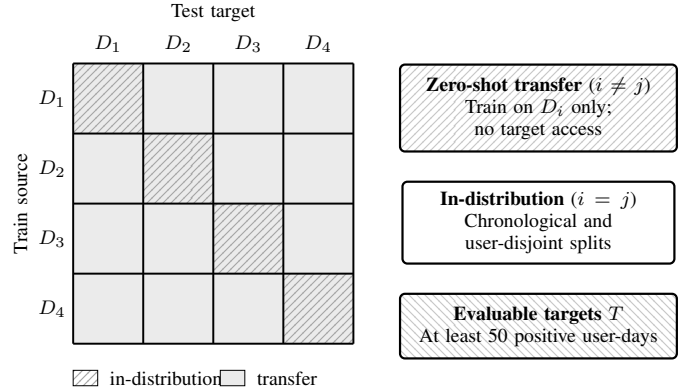


Fig. 3. Zero-shot evaluation protocol. Rows are training sources and columns are test targets; only evaluable targets enter the generalization gap.

The evaluation protocol is summarised in Fig. 3.

IV. METHODOLOGY

A. Heterogeneous Temporal Graph Construction

Each dataset is transformed into a heterogeneous temporal graph containing user and host nodes. Every security event creates a directed, action-typed edge between its associated user and host:

$$e = (u_e, h_e, a_e, t_e), \quad a_e \in \mathcal{A}. \quad (9)$$

The vocabulary $\mathcal{A}$ contains sixteen canonical actions: lo-gon, logoff, auth, file_read, file_write, file_delete, usb_connect, usb_disconnect, http, email_send, email_rcv, cmd_exec, ssh, ftp, process_start, and unknown. It was chosen to cover the activity types that the evaluated sources record in comparable form, namely session, authentication, file, removable-device, web, e-mail, and command events, without relying on dataset-specific content. The mapping from raw log fields to $\mathcal{A}$ is a static rule table in the data loaders. It does not use labels or model outputs, and the same tables were applied in all reported experiments. Dataset-specific event names that express the same behavioural operation are mapped to a shared action, whereas unmatched events are assigned to the reserved unknown action rather than discarded. Table I lists the rules, and Table II reports the resulting event distribution. No event in any domain was assigned to the unknown action. The distributions nevertheless differ markedly: LANL occupies only three actions, SPEDIA is dominated by command execution, and CERT is dominated by e-mail and web activity. The mapping therefore provides a common representation, but it cannot make the source domains semantically identical.

In addition to event edges, each user-day has an aligned feature vector

$$\mathbf{x}_{u,d} \in \mathbb{R}^F, \quad (10)$$

TABLE I. CANONICAL ACTION MAPPING. CERT R4.2 HAS NO ACTIVITY COLUMN FOR FILE AND E-MAIL EVENTS, SO ALL ITS FILE EVENTS MAP TO FILE_WRITE AND ALL ITS E-MAIL EVENTS TO EMAIL_SEND. CERT WEB EVENTS ARE SAMPLED INDEPENDENTLY WITH PROBABILITY 0.05 FOR EVERY USER, AND ELIGIBLE LANL AUTHENTICATION EVENTS WITH PROBABILITY 0.02 FOR EVERY USER (FIXED SEED 7; MACHINE ACCOUNTS REMOVED). GROUND-TRUTH ROSTERS ARE NOT USED IN THE SAMPLING DECISION, AND LABELS ARE ATTACHED ONLY AFTER SAMPLING. THE SSH, FTP, AND PROCESS_START ACTIONS ARE RESERVED AND ARE NOT PRODUCED BY ANY EVALUATED LOADER

Canonical action	CERT (r4.2/r5.2/r6.2)	SPEDIA (real-only)	LANL (auth.txt)
logon	logon.csv: Logon	Activity = session, other actions	Orientation LogOn
logoff	logon.csv: Logoff	Activity = session, action contains “logoff”	Orientation LogOff
auth	–	Activity = session, action contains “fail”	All other orientations
file_read	file.csv: File Open (r5.2/r6.2)	Activity = file, other actions	–
file_write	file.csv: File Copy, File Write; all r4.2 file events	Activity = file, action contains “modif” or “added”	–
file_delete	file.csv: File Delete	Activity = file, action contains “delet”	–
usb_connect	device.csv: Connect	Activity = device, other actions	–
usb_disconnect	device.csv: Disconnect	Activity = device, action contains “disconnect”	–
http	http.csv (sampled)	Activity = http	–
email_send	email.csv: Send; all r4.2 e-mail events	Activity = email	–
email_rcv	email.csv: View, Receive (r5.2/r6.2)	–	–
cmd_exec	–	Activity = command	–
unknown	Any unmatched token	Any other activity	–

TABLE II. DISTRIBUTION OF RETAINED EVENTS OVER CANONICAL ACTIONS AFTER LABEL-INDEPENDENT SAMPLING: COUNT (PERCENTAGE OF THE DOMAIN’S EVENTS). ONLY ACTIONS OBSERVED IN AT LEAST ONE DOMAIN ARE LISTED; SSH, FTP, AND PROCESS_START HAVE ZERO EVENTS IN EVERY DOMAIN. THE LAST ROW GIVES USER-DAYS AND, IN PARENTHESES, POSITIVE USER-DAYS

Action	CERT r4.2	CERT r5.2	CERT r6.2	SPEDIA	LANL
logon	470,591 (8.2%)	995,082 (4.2%)	1,948,933 (8.1%)	3,031 (14.7%)	3,786,729 (45.3%)
logoff	384,268 (6.7%)	814,988 (3.4%)	1,581,352 (6.6%)	2,038 (9.9%)	3,048,812 (36.4%)
auth	0 (0.0%)	0 (0.0%)	0 (0.0%)	330 (1.6%)	1,529,753 (18.3%)
file_read	0 (0.0%)	483,655 (2.0%)	657,500 (2.7%)	0 (0.0%)	0 (0.0%)
file_write	445,581 (7.7%)	403,966 (1.7%)	976,584 (4.1%)	4,027 (19.5%)	0 (0.0%)
file_delete	0 (0.0%)	0 (0.0%)	380,799 (1.6%)	712 (3.5%)	0 (0.0%)
usb_connect	203,339 (3.5%)	419,734 (1.8%)	778,354 (3.3%)	0 (0.0%)	0 (0.0%)
usb_disconnect	202,041 (3.5%)	417,250 (1.7%)	773,474 (3.2%)	0 (0.0%)	0 (0.0%)
http	1,420,426 (24.7%)	2,947,136 (12.4%)	5,850,511 (24.4%)	367 (1.8%)	0 (0.0%)
email_send	2,629,979 (45.7%)	5,555,187 (23.3%)	3,832,466 (16.0%)	0 (0.0%)	0 (0.0%)
email_rcv	0 (0.0%)	11,806,388 (49.5%)	7,162,491 (29.9%)	0 (0.0%)	0 (0.0%)
cmd_exec	0 (0.0%)	0 (0.0%)	0 (0.0%)	10,114 (49.1%)	0 (0.0%)
unknown	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Total events	5,756,225	23,843,386	23,942,464	20,619	8,365,294
User-days (positive)	330,452 (966)	693,649 (1,307)	1,394,010 (44)	256 (82)	389,323 (158)

whose dimensions have the same operational meaning across all datasets. The aligned features contain counts, binary activity indicators, timing variables, and other quantities that can be computed consistently from the common event ontology. Dataset identifiers, raw user names, host names, and source-specific record identifiers are excluded.

After processing all events up to the end of day d , the user-day representation is

$$\mathbf{r}_{u,d} = [\mathbf{s}_u(t_d^{\text{end}}) \parallel \mathbf{x}_{u,d}], \quad (11)$$

where, $\mathbf{s}_u(t_d^{\text{end}})$ is the temporal memory of user u and $\parallel$ denotes concatenation. A prediction head produces

$$\hat{y}_{u,d} = \sigma(\mathbf{W}_y \mathbf{r}_{u,d} + \mathbf{b}_y). \quad (12)$$

Fig. 4 shows the architecture.

B. Per-User Deviation Features

In line with user-centred and behavioural-baseline approaches [2], [16], insider activity is modelled relative to each user’s own history rather than solely through absolute activity counts. The same number of logons, file operations, or host accesses may therefore be routine for one user but unusual for another. To preserve temporal validity, the behavioural baseline is constructed causally from strictly earlier observations only. The transformation is also dataset-agnostic: it uses aligned numerical activity features and excludes dataset-specific tokens, URLs, email content, identifiers, or other source-dependent fields.

For user u , day d , and base feature x , the causal deviation feature is

$$\text{dev}_x(u, d) = \frac{x_{u,d} - \mu_{u,<d}}{\sigma_{u,<d} + \epsilon}, \quad (13)$$

where, $\mu_{u,<d}$ and $\sigma_{u,<d}$ are the mean and standard deviation of feature x over the strictly earlier days of user u , and

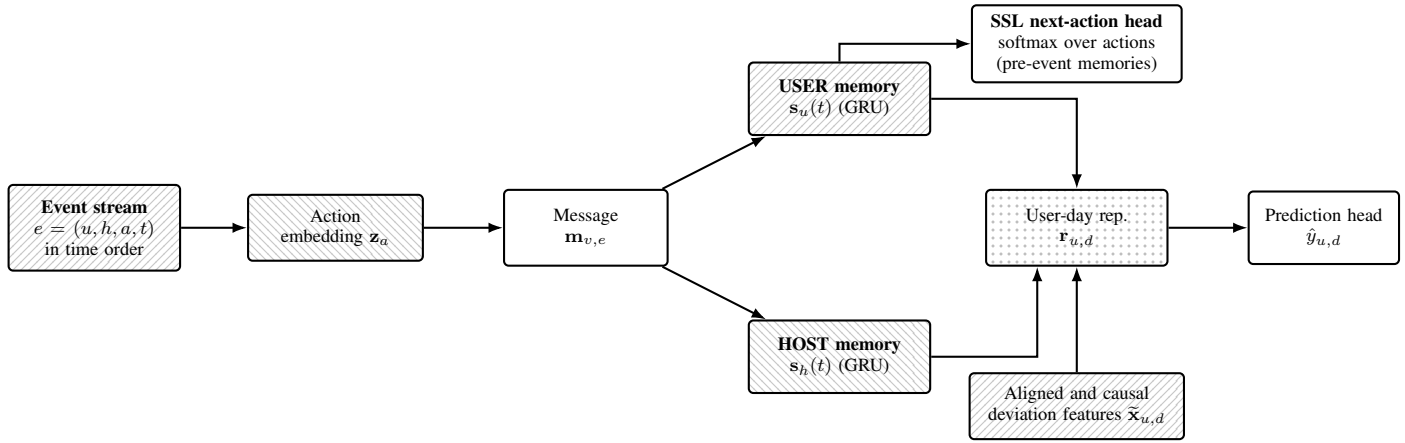


Fig. 4. Temporal graph detector used in the reported experiments. Events update user and host memories in chronological order; no explicit time encoding is used.

$\epsilon > 0$ prevents division by zero. Formally,

$$\mu_{u,<d} = \frac{1}{|\mathcal{H}_{u,d}|} \sum_{d' \in \mathcal{H}_{u,d}} x_{u,d'}, \quad \mathcal{H}_{u,d} = \{d' \mid d' < d\}, \quad (14)$$

with $\sigma_{u,<d}$ computed over the same historical set $\mathcal{H}_{u,d}$. When $\mathcal{H}_{u,d}$ is empty, corresponding to the first observed day of user u , $\text{dev}_x(u, d)$ is set to zero. The original base counts are retained, and the causal deviation features are appended. If the aligned base vector is $\mathbf{x}_{u,d} \in \mathbb{R}^F$, the resulting user-day feature vector is

$$\tilde{\mathbf{x}}_{u,d} = [\mathbf{x}_{u,d} \parallel \text{dev}(\mathbf{x}_{u,d})] \in \mathbb{R}^{2F}. \quad (15)$$

Because the same aligned base features and the same causal transformation are applied in every domain, the resulting feature space is identical across CERT, SPEDIA, and LANL.

1) *Leakage controls*: The following rules determine what information can reach each prediction. First, event sampling is label-independent and precedes label attachment (Section III), so the retained event volume of a user does not depend on whether that user is an insider. Second, each deviation value uses an expanding mean and standard deviation over the same user's strictly earlier days; the current day is excluded by a one-day shift, and a user's first observed day receives zero. Only activity counts and timing variables enter this computation, so labels never affect the history. Third, the features are computed once over each dataset's chronological event stream. Under the chronological split, a test day may therefore draw on the same user's training-period days, which precede it in time, but never on later days. Under the user-disjoint split, train and test users do not overlap, so a test user's history consists only of that user's own events. Fourth, during cross-dataset inference, target deviations are computed only from the same target user's earlier, unlabelled target events. Fifth, the temporal-graph memory is reset before each training epoch, before validation, and before scoring, so memory state is not carried from the training partition into the test partition or from the source into the target. Sixth, the temporal graph detector uses

no timestamp statistic of the scored stream, because it has no explicit time encoding (Section IV). Finally, standardisation statistics are fitted on the source training users only, excluding the validation users, and are applied unchanged to validation, test, and target data. Consequently, no feature value depends on future events, on labels, or on statistics of a test or target partition, and target labels do not enter model training or preprocessing.

2) *Shared feature-space limitation*: The shared, content-free feature space is a deliberate cross-dataset fairness constraint. It prevents models from exploiting dataset-specific tokens, URLs, email content, identifiers, or logging conventions that could artificially inflate within-dataset performance and reintroduce the artefacts examined by the benchmark. This restriction may reduce sensitivity to scenarios whose malicious meaning depends on content, such as particular URLs, email text, file names, or message semantics. Per-user deviation features partly mitigate this limitation by representing unusual behaviour relative to each user's own history, but cannot recover information absent from the aligned feature space.

C. Memory-Based Temporal Graph Backbone

The temporal backbone maintains a state vector $\mathbf{s}_v(t)$ for every user or host node v . For event $e = (u_e, h_e, a_e, t_e)$, a message is constructed separately for each endpoint $v \in \{u_e, h_e\}$:

$$\mathbf{m}_{v,e} = \mathbf{W}_m [\mathbf{s}_v(t_e^-) \parallel \mathbf{z}_{a_e}], \quad (16)$$

where, $\mathbf{z}_{a_e}$ is the embedding of action a_e , $\mathbf{s}_v(t_e^-)$ is the memory of node v immediately before event e , and separate projections $\mathbf{W}_m$ are used for user and host endpoints. Temporal graph models often add explicit temporal information, e.g., elapsed-time terms or functional time encodings [17], [18]. Here, such an encoding would need timestamp normalisation, and normalising with the scored target stream's first and last timestamps would use target-derived statistics. The reported configuration therefore uses no explicit time encoding. Temporal information enters the model through the chronological order in which events update the memories.

The node memory is updated as

$$\mathbf{s}_v(t_e) = \text{GRU}(\mathbf{m}_{v,e}, \mathbf{s}_v(t_e^-)). \quad (17)$$

Events are processed in chronological order. Consequently, the state used to predict event e contains only information available before t_e , preventing future-event leakage. Host memories allow activity on a shared machine to influence subsequent user representations, while user memories retain behavioural continuity across events and days.

D. Causal Self-Supervised Next-Action Prediction

The primary self-supervised task predicts the action associated with the next chronological event from the pre-event memories of its user and host. Let

$$\mathbf{g}(\mathbf{s}_{u_e}(t_e^-), \mathbf{s}_{h_e}(t_e^-)) = \text{softmax}(\mathbf{W}_g [\mathbf{s}_{u_e}(t_e^-) \parallel \mathbf{s}_{h_e}(t_e^-)] + \mathbf{b}_g) \quad (18)$$

be a probability distribution over the $A = |\mathcal{A}|$ action classes. The self-supervised loss is

$$\mathcal{L}_{\text{ssl}} = - \sum_{e \in \mathcal{E}_{\text{train}}} \log g_{a_e}(\mathbf{s}_{u_e}(t_e^-), \mathbf{s}_{h_e}(t_e^-)). \quad (19)$$

The loss is computed over all permitted training events rather than over a masked subset. The use of pre-event memories makes the objective causal because action a_e is not inserted into the state before its prediction. Self-supervised pre-training is conducted without malicious labels and is followed by supervised source-domain optimisation. A masked-action reconstruction variant was not evaluated in this study.

E. Supervised User-Day Detection

For labelled source-domain user-days, the prediction loss is binary cross-entropy:

$$\mathcal{L}_y = - \sum_{(u,d) \in \mathcal{S}^L} [y_{u,d} \log \hat{y}_{u,d} + (1 - y_{u,d}) \log (1 - \hat{y}_{u,d})]. \quad (20)$$

Any weighting or sampling used to address class imbalance is estimated from source-domain training data only. Target labels are never used to tune the classifier or its operating threshold. For zero-shot experiments, all components are trained with source data only. The temporal encoder is trained with the self-supervised objective and then frozen, after which the projection layer and prediction head are trained on labelled source user-days. Early stopping uses the validation PR-AUC of a held-out 15% of the source training users. The resulting model is frozen before evaluation on a target domain.

TABLE III. CONFIGURATION OF THE REPORTED EXPERIMENTS. WHERE TWO VALUES ARE GIVEN, THE FIRST APPLIES TO CERT SOURCES AND THE SECOND TO THE SPEDIA SOURCE

Item	Setting
Seeds	0–4 (splits, validation users, initialisation)
Event sampling	CERT web $p = 0.05$ and LANL authentication $p = 0.02$ for every user; seed 7; labels attached after sampling
Random Forest	300 trees; balanced-subsample class weights; unscaled features
Temporal memory	64-dimensional user and host GRU memories, updated in chronological event order
Embeddings	16-dimensional action embedding; no explicit time encoding
Readout	Linear-ReLU projection; MLP 128–64–1; dropout 0.1
Optimiser	AdamW; learning rate 3×10^{-4} ; weight decay 10^{-4} ; cosine schedule; gradient-norm clipping at 1.0
Epochs	at most 60 / 80; self-supervised pretraining 1 / 2 epochs
Batching	chronological edge batches of 16,384 / 8,192 events; one supervised update per day
Early stopping	validation PR-AUC on 15% of source training users; patience 8 / 12; minimum 12 / 20 epochs
Class weighting	positive weight $n_{\text{neg}}/n_{\text{pos}}$, capped at 100
Determinism	deterministic PyTorch algorithms; deterministic cuDNN, benchmarking off; CUBLAS workspace : 4096 : 8; hash seed 0
Bootstrap	5,000 resamples of paired per-seed differences; 95% percentile interval
Software	Python 3.11.15; scikit-learn 1.9.0; PyTorch 2.12.1 with CUDA 13.0; NumPy 2.4.4; pandas 3.0.3; SciPy 1.17.1
Hardware	NVIDIA GeForce RTX 5070 Laptop GPU (8 GB); 31 GB RAM

V. EXPERIMENTAL SETUP

A. Implementation Details

The pipeline is implemented in Python. The Random Forest uses scikit-learn, and the temporal graph detector uses PyTorch. Table III lists the configuration of both models. Class imbalance is addressed through positive-class weighting estimated only from source training data. The temporal graph detector standardises base features on the source training users and clips them and the causal deviation features (Section IV-B) to $[-8, 8]$; the Random Forest uses unscaled features. Every reported configuration was run with five random seeds (0–4); GPU training was deterministic. Tables IV and V report seed means, and paired seed-level bootstrap confidence intervals are reported for the two model comparisons in Table VI.

B. Cross-Dataset Protocol

The protocol trains on one source domain and evaluates on target domains without target access. All preprocessing, normalisation, feature selection, threshold selection, and model selection follow the access restrictions defined in Section III. Off-diagonal cells train on the complete source domain and score the complete target domain. For every evaluable domain, the in-distribution controls comprise a chronological split and a user-disjoint split. The chronological split places the earliest 70% of the timeline in training and the remainder in testing; the cut is taken at the 0.7 quantile of user-days for the Random Forest and of event timestamps for the temporal graph detector, so users can appear on both sides. The user-disjoint split assigns 70% of users to training and the remaining users to testing, stratified by whether a user has any positive user-day.

These controls test whether a high diagonal score depends on temporal proximity or on repeated users.

C. Compared Models

The Random Forest is the tabular reference model. It is evaluated on the complete source-by-target matrix and on both in-distribution splits. The temporal graph detector of Section IV is evaluated on the in-distribution controls of CERT r4.2, CERT r5.2, and SPEDIA and on the transfers from CERT r4.2 and CERT r5.2 into SPEDIA. Its complete transfer matrix was not computed because of its computational cost. Both models use the same user-day labels, aligned features, source-target partitions, and access restrictions.

D. Evaluation Metrics

PR-AUC (average precision) is the headline metric because positive user-days are rare. Because PR-AUC depends on prevalence, each value is also reported as base-rate lift, that is, PR-AUC divided by the positive rate of the evaluated set. Model differences are summarised by the mean paired difference across the five seeds, with a 95% percentile bootstrap interval from 5,000 resamples of the per-seed differences. Because the temporal graph detector was evaluated on two transfer cells only, no aggregate significance test across the transfer matrix is reported. The generalization gap in Eq. (8) is reported as a transparency statistic. Threshold-dependent operating-point metrics, such as the detection rate at a fixed false-positive rate or precision at k , are not used as evidence: the implementation derives their thresholds from the evaluated set, which would not be available in deployment.

VI. RESULTS AND DISCUSSION

All results use the label-independent sampling and the zero-shot protocol defined in Sections III–V, and all values are means over five seeds. This section first reports the in-distribution controls, then the Random Forest transfer matrix and the generalization gap, and finally the paired comparison between the temporal graph detector and the Random Forest.

A. In-Distribution Controls

Table IV shows that within-domain performance was heterogeneous. Under the user-disjoint split, the Random Forest reached PR-AUC values of 0.550 on CERT r4.2, 0.625 on CERT r5.2, and 0.989 on SPEDIA, but only 0.004 on LANL, whose test base rate was about 0.0004 (lift 9.0 $\times$). On both CERT releases, the chronological split gave higher scores than the user-disjoint split, for the Random Forest (0.666 and 0.815) and for the temporal graph detector (0.621 and 0.755). This pattern is consistent with a benefit from observing the same users during training. On SPEDIA, the Random Forest scored highly under both splits (0.989 and 0.992), whereas the temporal graph detector scored 0.807 under the user-disjoint split and 0.972 under the chronological split. The temporal graph detector scored below the Random Forest in every in-distribution cell. The chronological LANL split was not evaluable because its test period contained no positive user-days, and CERT r6.2 does not meet the evaluability criterion.

TABLE IV. IN-DISTRIBUTION PR-AUC (MEAN OVER FIVE SEEDS). “USER-DISJOINT” AND “CHRONOLOGICAL” DENOTE THE SPLIT; “TEMPORAL-GNN” DENOTES THE MODEL. N/E: NOT EVALUABLE (THE LANL CHRONOLOGICAL TEST PERIOD CONTAINS NO POSITIVE USER-DAYS; CERT R6.2 HAS 44 POSITIVE USER-DAYS, BELOW THE CRITERION OF 50). N/R: NOT RUN. USER-DISJOINT TEST BASE RATES ARE ABOUT 0.0027 (CERT R4.2), 0.0020 (CERT R5.2), 0.41 (SPEDIA), AND 0.0004 (LANL)

Domain	Random Forest		Temporal-GNN	
	User-disjoint	Chronological	User-disjoint	Chronological
CERT r4.2	0.5497	0.6660	0.4443	0.6211
CERT r5.2	0.6254	0.8150	0.4776	0.7554
SPEDIA	0.9895	0.9924	0.8072	0.9720
LANL	0.0036	n/e	n/r	n/e
CERT r6.2	n/e	n/e	n/e	n/e

B. Zero-Shot Cross-Dataset Transfer (C1)

Table V and Fig. 5 report the Random Forest transfer matrix. Transfer performance was substantially lower than in-distribution performance on average. Across the four evaluable targets, the mean in-distribution PR-AUC was 0.542 and the mean of the 16 evaluable zero-shot cells was 0.133, giving a generalization gap of $\Delta = 0.409$ (Eq. (8)). The decline was not uniform. Transfers between CERT releases retained part of the signal relative to the target base rate: these cells reached PR-AUC values between 0.015 and 0.390, with lifts between 5 $\times$ and 133 $\times$, although each remained below the in-distribution value of its target. In contrast, transfer into SPEDIA gave PR-AUC values between 0.302 and 0.363 against a target base rate of 0.32, corresponding to lifts of only 0.94–1.13 $\times$; the Random Forest therefore ranked SPEDIA user-days close to the level expected by chance. Transfers into LANL (PR-AUC ≤ 0.0010 ; lift 1.1–2.4 $\times$) and from LANL or SPEDIA into CERT (PR-AUC ≤ 0.0088 ; lift 1.0–3.0 $\times$) remained near the performance floor. Because the aggregate gap combines targets whose base rates differ by about three orders of magnitude, it is reported together with the per-cell values and lifts.

TABLE V. ZERO-SHOT TRANSFER PR-AUC OF THE RANDOM FOREST REFERENCE MODEL (BASE-RATE LIFT IN PARENTHESES); ROWS = TRAIN SOURCE, COLUMNS = TEST TARGET. THE DIAGONAL IS THE USER-DISJOINT IN-DISTRIBUTION SCORE (TABLE IV). CERT R6.2 IS A SOURCE ONLY; N/E = NOT EVALUABLE AS A TARGET

Train $\downarrow$ / Test $\rightarrow$	Target				
	r4.2	r5.2	r6.2	SPEDIA	LANL
r4.2	0.550 (207 $\times$)	0.250 (133 $\times$)	n/e	0.315 (0.98 $\times$)	0.0008 (2.0 $\times$)
r5.2	0.390 (133 $\times$)	0.625 (317 $\times$)	n/e	0.363 (1.1 $\times$)	0.0010 (2.4 $\times$)
r6.2	0.015 (5.1 $\times$)	0.135 (72 $\times$)	–	0.302 (0.94 $\times$)	0.0004 (1.1 $\times$)
SPEDIA	0.0088 (3.0 $\times$)	0.0037 (1.9 $\times$)	n/e	0.989 (2.7 $\times$)	0.0008 (1.9 $\times$)
LANL	0.0029 (0.99 $\times$)	0.0019 (1.0 $\times$)	n/e	0.341 (1.1 $\times$)	0.0036 (9.0 $\times$)

C. Temporal Graph vs. Tabular Representations (C2)

Table VI compares the temporal graph detector with the Random Forest on the two transfers into SPEDIA, using paired seed-level bootstrap intervals. The representation effect was source-dependent. For CERT r4.2 $\rightarrow$ SPEDIA, the temporal graph detector reached a mean PR-AUC of 0.438 (lift 1.37 $\times$), compared with 0.315 for the Random Forest, a paired difference of +0.123 (95% CI [+0.075, +0.169]); all five

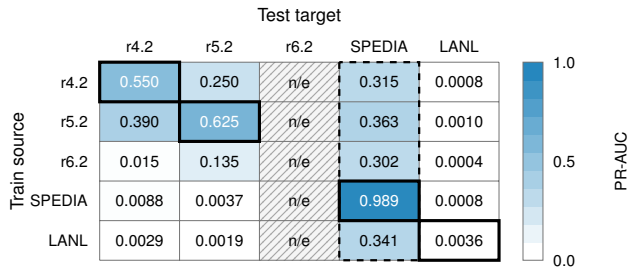


Fig. 5. Zero-shot cross-dataset transfer of the Random Forest (PR-AUC, mean of five seeds). Rows are training sources and columns are test targets; r4.2, r5.2, and r6.2 denote CERT releases. Outlined cells are user-disjoint in-distribution scores. The dashed SPEDIA column has a base rate of 0.32, so its transfer values of 0.30–0.36 are close to chance. CERT r6.2 is not evaluable as a target (hatched) and is excluded from the generalization gap; it is retained as a training source.

seeds favoured the graph model. For CERT r5.2→SPEDIA, the Random Forest reached 0.363 and the temporal graph detector 0.332 (lift 1.04×), a paired difference of -0.031 (95% CI $[-0.061, -0.012]$); all five seeds favoured the Random Forest. The graph advantage observed from CERT r4.2 was therefore not reproduced from CERT r5.2, although both sources come from the same synthetic generator and the target is identical. Because the temporal graph detector was not evaluated on the remaining transfer cells, these two comparisons do not support a general statement about graph and tabular representations across the benchmark.

TABLE VI. PAIRED COMPARISON OF THE TEMPORAL-GNN (NO EXPLICIT TIME ENCODING) AND THE RANDOM FOREST ON TRANSFERS INTO SPEDIA (TARGET BASE RATE 0.32). Δ PR = PR-AUC(TEMPORAL-GNN) – PR-AUC(RF), AVERAGED OVER FIVE PAIRED SEEDS; THE 95% CI IS A PERCENTILE BOOTSTRAP OVER THE PAIRED PER-SEED DIFFERENCES (5,000 RESAMPLES). THE INTERPRETATION NAMES THE MODEL FAVOURED BY AN INTERVAL THAT EXCLUDES ZERO

Source → Target	RF	Temporal-GNN	Δ PR	95% CI	Interpretation
CERT r4.2 → SPEDIA	0.315	0.438	+0.123	[+0.075, +0.169]	GNN supported
CERT r5.2 → SPEDIA	0.363	0.332	-0.031	[-0.061, -0.012]	RF supported

D. Discussion

The results show that strong single-dataset performance is not evidence of cross-dataset transfer. Even where in-distribution scores were high, as for SPEDIA, zero-shot transfer from the other domains into the same target ranked positive user-days close to chance for the Random Forest. The only transfer directions with substantial base-rate lift were those between CERT releases produced by the same simulator. The in-distribution controls add a further caution. On both CERT releases, chronological splits produced higher PR-AUC than user-disjoint splits, so a within-dataset evaluation that places the same users in training and testing can overstate performance on unseen users.

The paired comparison indicates that representation effects depend on the source. The temporal graph detector improved on the tabular reference when trained on CERT r4.2 but not when trained on CERT r5.2, and it scored below the reference in every in-distribution control. A graph representation

can therefore help for a particular source–target pair without providing a consistent advantage, even when the target is fixed. Because the complete graph transfer matrix was not computed, the benchmark does not establish whether graph representations are generally better or worse than tabular ones.

Transfer involving LANL remained near the performance floor in both directions. Three non-exclusive explanations are consistent with the evidence. First, the threat semantics differ: LANL positives mark red-team authentication and lateral-movement activity, whereas CERT positives mark synthetic traitor scenarios and SPEDIA positives follow descriptive and Wazuh severity rules. Second, the shared representation covers LANL only partially. All 8.37 million sampled LANL events map to three of the sixteen canonical actions (logon 45.3%, logoff 36.4%, and auth 18.3%; Table II). Authentication type, logon type, and outcome are therefore collapsed, and the file, device, web, and e-mail actions that dominate CERT are absent. Third, prevalence differs by orders of magnitude: the positive user-day rate is about 0.0004 for LANL (158 of 389,323 user-days), compared with 0.32 for SPEDIA and between about 0.00003 and 0.003 for the CERT releases. The present benchmark cannot determine how much of the LANL failure is attributable to each mechanism, and none of them is claimed to be the sole cause.

Finally, the reported results concern ranking and discrimination transfer rather than probability calibration transfer. PR-AUC measures how well positive user-days are ranked above negative ones under class imbalance. It does not show that predicted probabilities keep the same meaning in a new domain, that source confidence scores remain calibrated on the target, or that a fixed operating threshold preserves its false-positive rate. No target recalibration was performed, and calibration metrics such as the Brier score or expected calibration error were not computed. Calibration transfer and operating-point stability therefore remain open questions for deployment.

VII. THREATS TO VALIDITY

The datasets differ in collection method, event coverage, label semantics, and insider type. CERT contains synthetic traitor scenarios, LANL labels red-team authentication activity, and SPEDIA uses descriptive and Wazuh severity rules. Mapping these sources to a common user-day label supports comparison but cannot make their underlying threat concepts identical. The study therefore interprets cross-dataset results as transfer across related security domains rather than as proof of universal insider-threat detection.

SPEDIA contains CERT-derived records in its original release. This risk is mitigated by excluding all rows for which `Decoder_name == 'cert'` and auditing the remainder for CERT-related tokens and identifiers. The audit supports a clean comparison, but it cannot establish that every retained behavioural pattern is statistically independent of all external data-generation influences.

The fixed action vocabulary and aligned feature set may omit useful dataset-specific information. Conversely, retaining source-specific identifiers or unsupported fields could allow models to recognise the source dataset rather than the behaviour. The study therefore reports the mapping rules

and excludes direct dataset identifiers. No alternative action groupings were evaluated. Because no event in any domain was assigned to the unknown action, retaining or excluding unknown events does not change the reported results; however, sensitivity to alternative groupings, such as separating authentication subtypes in LANL, remains untested. The benchmark compares one tabular and one temporal graph model; other detector families were not evaluated under the final protocol, and the temporal graph detector was evaluated on two transfer cells only.

LANL red-team activity is concentrated within a limited sub-window of the collection period. A chronological split therefore leaves no positives in the test period, so the LANL in-distribution diagonal is reported under the user-disjoint split only. LANL timestamps are also expressed on a relative clock, so time-of-day features are interpreted through the per-user deviation transformation rather than as absolute working hours. The generalization gap assigns uniform weight to all diagonal and off-diagonal cells. This keeps the statistic transparent, but it does not account for differences in target size, prevalence, or label uncertainty. Because target prevalence ranges over roughly three orders of magnitude across domains, per-cell PR-AUC and base-rate lift are reported alongside the aggregate gap, and CERT r6.2 (44 positive user-days) is excluded from the evaluable targets.

The confidence intervals are percentile bootstrap intervals over five paired seeds. They capture variation across seeds but not sampling variation in the evaluated user-days, and five seeds give only a coarse estimate of that variation. The CERT web logs and the LANL authentication log were subsampled to make processing feasible. The sampling rate is identical for all users and independent of labels, but it reduces the evidence available per user-day, and a labelled user-day whose events are all removed cannot be observed (18 of the 176 LANL red-team user-days). The reported values therefore refer to the sampled event universe rather than to the complete logs. Finally, the temporal graph detector was evaluated without explicit time encoding in order to exclude target-derived timestamp statistics. Configurations with a source-only or causal time encoding were not evaluated.

VIII. CONCLUSION AND FUTURE WORK

This study presents, to the best of our knowledge, the first systematic zero-shot cross-dataset benchmark for insider threat detection across CERT, SPEDIA and LANL. The benchmark uses label-independent event sampling, a common user-day prediction unit, chronological and user-disjoint in-distribution controls, and an evaluable-set generalization gap, and no labelled or unlabelled target observations are used during training. Within-domain performance was heterogeneous, and transfer performance was substantially lower on average. For the Random Forest reference model, mean PR-AUC fell from 0.542 in distribution to 0.133 under zero-shot transfer, a generalization gap of 0.409. Transfer into SPEDIA was close to the target base rate, and transfer into and out of LANL remained at the performance floor. A memory-based temporal graph detector without explicit time encoding outperformed the Random Forest for CERT r4.2→SPEDIA but underperformed it for CERT r5.2→SPEDIA. The representation effect was therefore source-dependent, and no general graph advantage

is claimed. Strong single-dataset performance is insufficient evidence of cross-dataset generalization.

Future work will extend the benchmark to further independently constructed corpora, including TWOS and DARPA OpTC, and will evaluate the temporal graph detector on the complete transfer matrix. It will also examine source-only or causal time encodings and target-assisted domain adaptation as separately reported settings, evaluate event-, session-, host-, and process-level prediction units, assess calibration transfer and operating-point stability, and test alternative action groupings. Future work will also examine whether attribution profiles remain stable across domains.

DECLARATIONS

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

Data Availability

This study uses only third-party, publicly documented datasets, which are not redistributed by the authors. Each is available from its original source under its respective licence: the CERT Insider Threat Test Dataset; the SPEDIA dataset (Zenodo, <https://doi.org/10.5281/zenodo.15495572>); and the LANL Comprehensive Multi-Source Cyber-Security Events dataset.

Code Availability

The code (including the configuration of the camera-ready experiments), benchmark documentation and summary result tables supporting the camera-ready experiments are publicly available at <https://github.com/alriyami7777-alt/crossdataset-insider-threat>. The third-party datasets are not redistributed and must be obtained from their original sources.

Author Contributions

QASIM Mohamed Muhanna Alriyami: Conceptualization, Methodology, Software, Investigation, Formal analysis, Data curation, Validation, Visualization, Writing – original draft, Writing – review and editing. Mohd Murtadha Bin Mohamad: Supervision, Methodology, Validation, Writing – review and editing, Project administration.

Declaration of Generative AI Use

Generative AI tools were used during manuscript preparation for drafting assistance, language refinement and presentation support. All scientific decisions, experimental results, interpretations and final manuscript content were critically reviewed and verified by the authors, who take full responsibility for the work.

Ethical Approval

This work did not involve human participants or animals and did not collect any new personal data. All experiments were conducted on previously published, de-identified cyber-security datasets; no ethical approval was therefore required.

REFERENCES

- [1] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," in *Proc. IEEE Security and Privacy Workshops*, 2013, pp. 98–104. <https://doi.org/10.1109/SPW.2013.37>
- [2] Y. Qi, C. Yan, Z. Wang, C. Zhang, S. Liu, Z. Lu, and B. Jiang, "ATHITD: Attention-based temporal heterogeneous graph neural network for insider threat detection," *Computers & Security*, vol. 157, p. 104587, 2025. <https://doi.org/10.1016/j.cose.2025.104587>
- [3] K. Fei, J. Zhou, L. Su, W. Wang, and Y. Chen, "Log2Graph: A graph convolution neural network based method for insider threat detection," *Journal of Computer Security*, vol. 33, no. 1, pp. 37–56, 2025. <https://doi.org/10.3233/JCS-230092>
- [4] X. Cai *et al.*, "LAN: Learning adaptive neighbors for real-time insider threat detection," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 10157–10172, 2024. <https://doi.org/10.1109/TIFS.2024.3488527>
- [5] P. Pal, P. Chattopadhyay, and M. Swarnkar, "Temporal feature aggregation with attention for insider threat detection from activity logs," *Expert Systems with Applications*, vol. 224, p. 119925, 2023. <https://doi.org/10.1016/j.eswa.2023.119925>
- [6] A. Alshehri, "Relational deep learning detection with multi-sequence representation for insider threats," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 758–765, 2022. <https://doi.org/10.14569/IJACSA.2022.0130587>
- [7] K. Kundiya and Y. Haribhakta, "A systematic review on insider threat detection using natural language processing," *International Journal of Information Security*, vol. 24, no. 6, art. 227, 2025. <https://doi.org/10.1007/s10207-025-01145-6>
- [8] S. Layeghy and M. Portmann, "On generalisability of machine learning-based network intrusion detection systems," *arXiv preprint*, 2022. <https://doi.org/10.48550/arXiv.2205.04112>
- [9] M. Cantone, C. Marrocco, and A. Bria, "On the cross-dataset generalization of machine learning for network intrusion detection," *arXiv preprint*, 2024. <https://doi.org/10.48550/arXiv.2402.10974>
- [10] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection," *Big Data Research*, vol. 30, p. 100359, 2022. <https://doi.org/10.1016/j.bdr.2022.100359>
- [11] O. Oladejo and A. A. Ahmed, "Leveraging cross-domain transfer learning for enhanced multi-protocol network intrusion detection," *Computers*, vol. 15, no. 6, art. 376, 2026. <https://doi.org/10.3390/computers15060376>
- [12] D. Álvarez, L. Pérez, A. Mateo, X. Larriva-Novo, M. Álvarez-Campana, and V. A. Villagrà, "SPEDIA dataset," *Zenodo*, 2025. <https://doi.org/10.5281/zenodo.15495572>
- [13] A. D. Kent, "Comprehensive, multi-source cyber-security events data set," Los Alamos National Laboratory, 2015. <https://doi.org/10.17021/1179829>
- [14] A. K. Paduvilan, S. A. Babu, J. Sekar, A. K. Elengovan, N. Seshagiri, K. Jangiti, and A. K. Selvam, "Attention-driven fusion of sequential and graph neural models for insider threat detection in UEBA," *Scientific Reports*, vol. 16, art. 21117, 2026. <https://doi.org/10.1038/s41598-026-41242-7>
- [15] A. Harilal, F. Toffalini, J. Castellanos, J. Guarnizo, I. Homoliak, and M. Ochoa, "TWOS: A dataset of malicious insider threat behavior based on a gamified competition," in *Proc. MIST@CCS*, 2017, pp. 45–56. <https://doi.org/10.1145/3139923.3139929>
- [16] D. C. Le and N. Zincir-Heywood, "Exploring anomalous behaviour detection and classification for insider threat identification," *Int. J. Network Management*, vol. 31, no. 4, p. e2109, 2021. <https://doi.org/10.1002/nem.2109>
- [17] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," in *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020. <https://doi.org/10.48550/arXiv.2006.10637>
- [18] D. Xu, C. Ruan, E. Körpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020. <https://doi.org/10.48550/arXiv.2002.07962>