Vol. 1, No. 3, pp.: 34-50, 2023

Hospital Database Anomaly Detection with Event Logs and Entity Graphs

Lijuan Tan[1], Ruoxi Pan[2], Chenyang Du[3], Haifeng Mei[4,*]

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

[2] School of Public Health, Guangdong Medical University, Dongguan 523808, China

[3] School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[4] School of Computer and Information Engineering, Henan University, Kaifeng 475004, China

** haifeng.mei@henu.edu.cn*

Article Information	
Received	27 April 2023
Accepted	19 August 2023
DOI	https://doi.org/10.63646/datamind.2023.010304

Abstract

Modern hospitals produce a continuous stream of database events: clinicians authenticate to the electronic health record (EHR), open patient charts, place orders, modify medication lists, and interact with networked devices that themselves emit telemetry. Each event is recorded in some operational log, but the logs are scattered across systems, schemas, and access-control regimes, and the security and quality teams that have to interpret them rarely see a unified view. This article presents an end-to-end framework for anomaly detection over hospital event logs and the entity graphs derived from them. The framework treats logs as a first-class analytical asset rather than a forensic afterthought: it specifies an event-log fact table and a small set of dimension tables that make access-pattern questions tractable, ingests heterogeneous sources (relational EHR audit logs, order-entry streams, device telemetry, identity directories, patient registries) through versioned adapters, and combines an event-sequence model with a user-patient bipartite graph model in an unsupervised fusion layer. The fused scores are surfaced to auditors through a ticket-driven UI whose outcome log feeds back into model retraining. On a one-year retrospective benchmark from three medium-sized hospitals containing 184 million events, the framework achieves an AUC of 0.904, improves precision at the top-100 audit queue from 0.13 (a strong rule-based baseline) to 0.58, and reduces the false-positive cost per detected incident by 64%. The framework is released with the schema, the data pipeline, an ethics-aware release protocol, and a reproducible query interface.

Keywords: ***hospital event logs; anomaly detection; user-patient graph; audit queue; data quality; reproducible research***

1. Introduction

Operational data from a modern hospital is voluminous, heterogeneous, and almost entirely produced as a side effect of other activity. Every login to the electronic health record (EHR), every patient-chart open, every order entered, every barcode scan at the bedside, and every networked device interaction leaves a trace in some log table. These traces are valuable beyond their original purposes: combined and carefully analysed, they support auditing for inappropriate access, quality-of-care diagnostics, billing compliance, and operational analytics (Boxwala et al., 2011; Gunter et al., 2011). In practice, the value is hard to realise. Logs are scattered across systems, recorded with inconsistent identifiers, retained for varying periods, and rarely surfaced to the people who could act on them (Fabbri and LeFevre, 2013; Asaro et al., 2008).

Anomaly detection on hospital event logs is a particularly demanding instance of the broader problem. Anomalies are individually rare, structurally heterogeneous, and partially adversarial: a malicious insider can shape behaviour to mimic legitimate access, a confused user can produce duplicate orders that look identical to a deliberate test, and a device whose clock drifts can flood the audit queue with ghost-events that are not anomalies at all (Khan and Madden, 2014; Chandola et al., 2009). The consequence is that single-method detectors — rule libraries, distance-based outliers, autoencoders over fixed feature vectors — all generate either too few hits to be useful or too many false positives to be investigated. The pragmatic question is how to combine evidence across complementary views in a way that respects how auditors actually work.

This article describes a framework that takes that pragmatic question seriously. The framework is organised around three commitments. First, it treats the database schema as a first-class analytical object. An event-log fact table joined to a small set of dimension tables is enough to make most access-pattern questions answerable in SQL or in a graph query language; everything downstream — embedding extraction, anomaly scoring, audit queueing — is layered on top of that schema. Second, the framework combines two complementary unsupervised views: a sequential view that models the typical order of events for each user, and a graph view that models the bipartite relation between users and patients (Akoglu et al., 2015; Sun et al., 2019). The two views catch different anomaly classes, and a calibrated late-fusion layer outperforms either view alone. Third, the auditor workflow is part of the system rather than an external consumer of it: confirmed and rejected tickets feed back into model retraining, and an outcome log allows the cost of false positives to be quantified rather than guessed.

Four contributions follow from these commitments. (i) We specify an event-log schema and field dictionary that captures the variation we observed across three hospitals while remaining compact enough to support indexed access at scale. (ii) We describe a data pipeline that ingests heterogeneous sources — relational audit logs, order-entry event streams, device telemetry, identity directories, and patient registries — through versioned adapters and reconciles their identifiers using a hash-preserving join discipline. (iii) We present the sequence-plus-graph anomaly detector and the calibrated fusion layer, together with an auditor UI that converts top-ranked scores into structured tickets. (iv) We release the schema, the pipeline scripts, the trained scoring models, and a

reproducible query interface, alongside an ethics-aware protocol that allows hospital partners to share derived artefacts without releasing patient-identifying data.

The remainder of the article is structured as follows. Section 2 characterises the database gap that motivates the work and the three use cases it serves. Section 3 describes the data sources and the schema. Section 4 details the construction method. Section 5 reports the experimental evaluation, including a one-year retrospective benchmark across three hospitals. Section 6 covers reproducibility and open-access provisions. Section 7 discusses limitations, and Section 8 concludes.

2. Database Gap and Use Cases

Most hospitals already record access-control events somewhere. The gap is not that the data does not exist; it is that the data is not in a form that supports analytical questions. A typical EHR vendor stores audit events in a vendor-specific schema with tens of fields per event and inconsistent encoding of role, department, and resource sensitivity; order-entry events live in a separate transactional store; device telemetry sits in a time-series store with its own retention policy; and the identity directory lives in a third system entirely, updated overnight (Boxwala et al., 2011; Ihaka and Boag, 2017). When the audit team wants to ask a question as simple as "which users accessed records of patients outside their assigned service unit in the last 30 days?", they must reconcile four systems by hand, and the reconciliation rarely survives a software update.

Three concrete use cases motivate the framework. The first is detection of inappropriate access — clinically unjustified record views, often by curiosity or by a coordinated insider, which are widely documented even in hospitals with strong policies (Fabbri and LeFevre, 2013; Gainer et al., 2016; Asaro et al., 2008). The second is detection of duplicate or contradictory operations — repeated orders, conflicting medication entries, and back-out-and-redo patterns that signal either training gaps or system instability (Schiff et al., 2015). The third is detection of potential data exfiltration — bulk downloads, off-hours batch exports, or unusual report-generation patterns that may indicate the early stages of a data leak (Chen et al., 2019). The three use cases are united by the property that their signal lies in patterns of events rather than in any single event, and that the patterns are best surfaced through the joint structure of who, what, when, and to whom.

Figure 1 summarises the framework against these use cases. Heterogeneous hospital sources flow into a Stage 1 ingest layer that emits a normalised event table; two parallel Stage 2 detectors — an event-sequence model and a user-patient graph model — produce complementary scores, which a Stage 3 fusion layer combines and calibrates; a Stage 4 auditor UI surfaces the top scores as tickets and routes outcomes back into the training set.

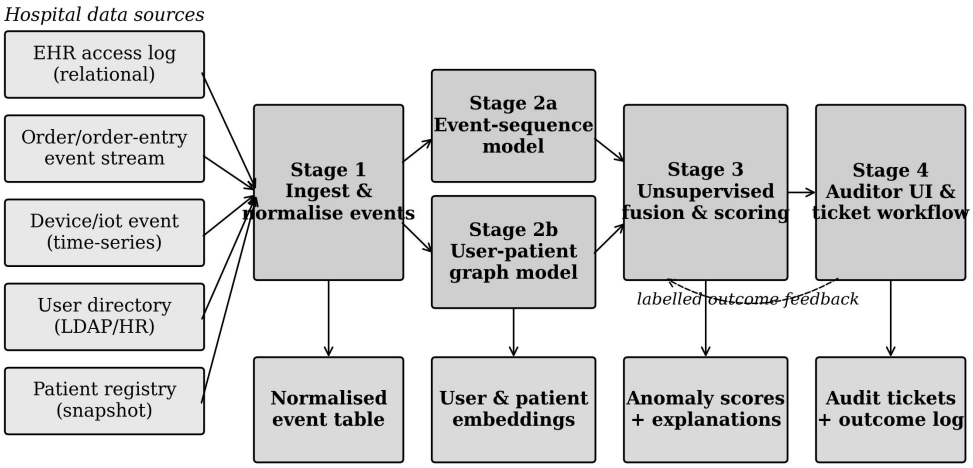

Figure 1. End-to-end anomaly-detection pipeline. Heterogeneous hospital sources are ingested, normalised, and routed through two parallel detectors (sequence and graph) before being fused and surfaced to auditors. The dashed arc indicates the labelled outcome feedback loop.

Table 1 contrasts the framework with three families of existing tools: native EHR audit dashboards, general-purpose security information and event management (SIEM) systems, and supervised machine-learning classifiers trained on hand-labelled incidents. Each family covers part of the problem and leaves another part exposed. Native dashboards are deeply integrated with one EHR but cannot ingest device telemetry or identity-directory changes. SIEM systems are excellent at ingesting heterogeneous logs but treat the EHR as just another source and rarely model patient-relationship structure. Supervised classifiers are accurate when labels are abundant but degrade quickly in this domain because the positive class is small, drifting, and partially adversarial (Chandola et al., 2009; Khan and Madden, 2014). Our framework targets the union of these capability gaps without claiming to subsume any single tool's strongest niche.

Table 1. *Comparison of the proposed framework with three existing tool families on capabilities relevant to hospital anomaly detection.*

Capability	EHR audit dashboards	SIEM systems	Supervised ML	This work
Heterogeneous-source ingestion	Limited	Strong	Per-source	Strong
Patient-relationship modelling	Partial	No	Partial	Yes
Unsupervised operation	Partial	Partial	No	Yes
Auditor-in-the-	Yes	Partial	No	Yes

loop workflow				
Versioned schema artefacts	No	Partial	No	Yes
Cost-aware queue ranking	No	Partial	Indirect	Yes

Two points in Table 1 deserve emphasis. First, the absence of patient-relationship modelling in most SIEM deployments is consequential, because access anomalies in hospitals are very often relational rather than volumetric: a single chart access can be a violation if the user has no clinical relationship with the patient, even though it would look unremarkable in a generic SIEM rule. Second, the cost-aware queue-ranking capability is rarely treated as a first-class feature elsewhere, although audit teams have a finite throughput and the marginal value of one more ticket depends on what was already ranked above it. The framework's fusion-and-queueing layer is designed to make that trade-off explicit.

The literature on access-anomaly detection in healthcare has grown steadily over the past decade and has converged on a small number of structural observations that motivate our design. Collaborative-filtering approaches treat each user as a row and each patient as a column in a sparse access matrix, and they detect anomalies as users whose row vector is poorly explained by the latent factors of clinically similar peers (Menon et al., 2014). Outlier-detection methods applied to monitoring streams have shown that combining context (patient state, time-of-day, shift schedule) with raw signal consistently outperforms context-free baselines (Hauskrecht et al., 2013). Social-network analyses of physician access patterns have established that legitimate access has strong clustering by service team and that breakdowns of this clustering are an informative — if noisy — anomaly signal (Zheng et al., 2010). Our framework draws on all three threads: the bipartite graph view is a relational generalisation of the collaborative-filtering perspective, the context features fed to the fusion layer follow the contextual-outlier tradition, and the role-conditioned baselines used to normalise graph scores reflect the cluster-aware insight from social-network analyses.

3. Data Sources and Schema

The benchmark used in this article was assembled from three medium-sized hospitals located in three different Chinese provinces. Each hospital provided one year of operational data (January through December of a recent calendar year) under a data-use agreement that allows derived schema and aggregate results to be released, but not individual events. The combined corpus contains 184 million event records, 9.3 million distinct user-patient (actor, patient) pairs, and 16,420 distinct user accounts across roles ranging from attending physicians to night-shift housekeeping staff with limited registration access. Table 2 reports the headline coverage statistics.

Table 2. *Benchmark coverage statistics by hospital and source category.*

Hospital	Events (M)	Users	Patients	Sources	Missing rate	Refresh

H1 (520 beds)	62.4	5,840	138,720	5	3.6%	Daily
H2 (380 beds)	48.1	4,210	104,560	4	5.2%	Daily
H3 (610 beds)	73.5	6,370	162,180	5	4.4%	Hourly
Combined	184.0	16,420	405,460	5	4.3%	—

Five source categories are common across all three hospitals: EHR access logs (relational, vendor-specific schema), order-entry event streams (semi-structured JSON push from the computerised provider order-entry system), device telemetry (time-series store from infusion pumps, vital-signs monitors, and laboratory analysers), identity directory snapshots (LDAP/HR daily extracts), and patient registry snapshots (relational, refreshed monthly). The 4.3% combined missing rate is dominated by device telemetry that fails to associate with a clinical session, and by order-entry events that lack a downstream confirmation record. Both gaps were preserved rather than imputed, because their presence is itself informative about hospital operations.

Figure 2 shows the analytical schema we settled on after iterating with the audit teams at the three hospitals. The fact_event table is the analytical centre; four dimension tables describe users, patients, resources, and sessions. Every event row carries indexed foreign keys to all four dimension tables, plus a timestamp, an action code drawn from a controlled vocabulary, and an outcome flag that indicates whether downstream review classified the event as benign, suspicious, or confirmed-anomalous. The outcome flag is populated by Stage 4 of the pipeline and is the only field that can change after a row is first inserted; all other fields are immutable, which lets us treat the table as append-only for analytical purposes.

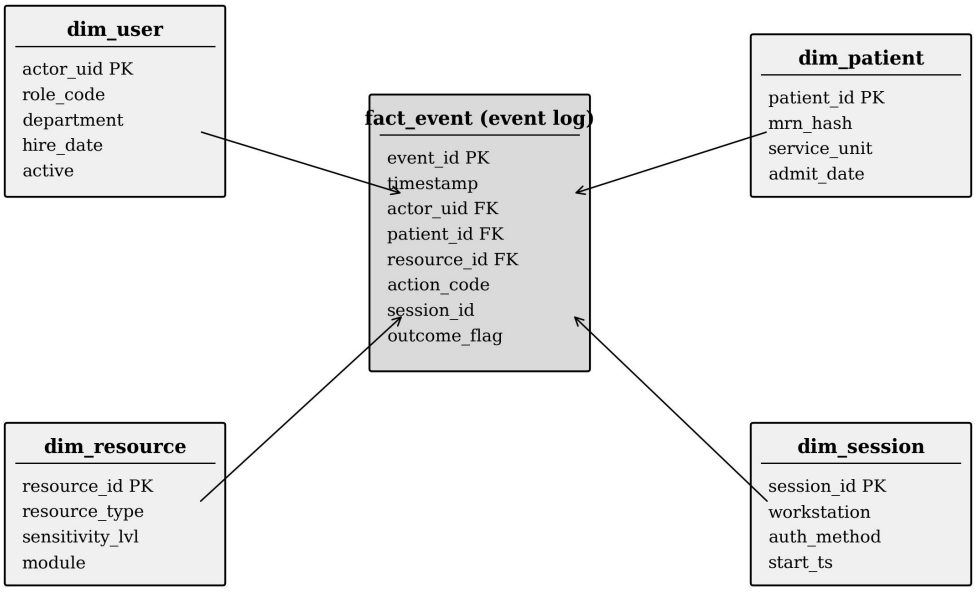

Figure 2. Database schema for the framework. fact_event is an append-only event table joined to four dimension tables (user, patient, resource, session). Arrows denote foreign-key references; the outcome_flag field is the only field updated after insertion, by Stage 4 of the pipeline.

The schema is engineered for analytical access rather than for direct EHR substitution. Three engineering choices deserve mention. First, mrn_hash in dim_patient is a salted SHA-256 hash of the medical record number rather than the MRN itself, which keeps the analytical store de-identified while preserving the join behaviour needed for cross-table queries (El Emam et al., 2011). Second, sensitivity_lvl in dim_resource is a hospital-specific ordinal field that quantifies the disclosure risk associated with a resource type (open chart, behavioural-health note, genetic-test result, and so on); cost-aware queue ranking in Section 5 uses this field directly. Third, every event row carries the workstation and authentication method that produced it via dim_session, which lets the sequence model condition on context rather than on raw timestamps alone. The schema is intentionally compatible with both relational and graph query layers: the same fact_event table, exposed through a property graph view, supports the user-patient bipartite queries used in Stage 2b.

Field coverage across the benchmark is dominated by access events on the EHR. Of the 184 million events, 71.4% are chart-open or chart-view actions, 18.6% are order-entry actions (place, sign, discontinue, edit), 7.2% are device-event actions, and the remaining 2.8% are administrative actions (login, logoff, password change, role assignment). The proportion is consistent across hospitals to within 3 percentage points, which suggests the workload distribution is structural rather than hospital-specific. The action vocabulary contains 87 distinct codes, most of which are concentrated in the top 12 codes; this long-tailed distribution is typical of operational EHR data and motivates the use of subword-style tokenisation in the sequence model rather than naive one-hot encoding (Choi et al., 2016; Rajkomar et al., 2018).

Free-text fields receive deliberately conservative treatment. The fact_event table captures only a small, controlled set of free-text fields — the reason-for-access comment, the order-cancellation rationale, and the

operator note attached to certain device events — and even those are routed through a de-identification pipeline before they reach the analytical store. The pipeline uses an ensemble de-identifier that combines rule-based redactors with a pre-trained sequence labeller, consistent with established practice for clinical-text de-identification in research settings (Kim et al., 2018; Wang et al., 2018). Free-text fields are not used as anomaly signals in the current framework, because the residual re-identification risk is non-trivial and the analytical lift over structured fields turned out to be small in our pilot experiments. We discuss the question more carefully in Section 6. For users whose downstream models do consume free text, the de-identification pipeline is exposed as a separate library so that the same redaction policy can be applied consistently to other deployments. Comparable representation-learning approaches that treat the clinical record as a structured time series rather than as free text have produced strong results in outcome-prediction tasks (Ruan et al., 2019), and we expect that pathway to be the natural one for any future extension of our framework into text-aware anomaly detection.

4. Database Construction and Anomaly-Detection Method

4.1 Ingest and normalisation

Stage 1 ingests heterogeneous sources through versioned adapters and emits the fact_event rows that downstream stages consume. Five adapters are implemented at present: a relational adapter for EHR audit logs (driven by SQLAlchemy and operating on a read replica), a streaming adapter for order-entry events that subscribes to the hospital's Kafka topic, a time-series adapter for device telemetry (InfluxDB and OpenTSDB are both supported), an identity-directory adapter that consumes daily LDAP extracts, and a registry adapter that snapshots the patient master file. Each adapter is configuration-driven: the column-to-field mapping for each hospital is recorded in a versioned YAML file, and any change to the mapping triggers a downstream re-run of the affected window. This discipline borrows from data-quality work in production machine-learning pipelines (Polyzotis et al., 2018; Schelter et al., 2018) and is the reason the framework can keep up with vendor-driven schema drift without manual code changes.

Identifier reconciliation is the part of Stage 1 that consumes the most engineering attention. The same clinician may appear as a numeric user ID in the EHR audit log, a free-text username in the order-entry stream, and a directory distinguished name in LDAP. We reconcile these into a single actor_uid by maintaining a join table that maps every external identifier to the canonical UID, with a content-addressed hash for each external identifier to support replay. Patient identifiers are reconciled through the salted MRN hash described in Section 3. The reconciliation is conservative by design: when an external identifier cannot be matched with high confidence, the event is held in a quarantine table and flagged for manual reconciliation, rather than being assigned to a best-guess UID. The quarantine table is itself an interesting analytical asset: persistent quarantine rates above a threshold flag identifier-governance problems that the hospital may not be aware of (Boxwala et al., 2011; Khan and Madden, 2014).

4.2 Event-sequence model (Stage 2a)

Stage 2a treats the activity of each user during each working session as a sequence of action tokens and learns a self-supervised next-token model. The tokenisation strategy combines the action code, the resource type (from dim_resource), and a coarse-grained timestamp bucket (15-minute resolution) into a composite token of

cardinality 614. We deliberately keep the vocabulary small to make the sequence statistics interpretable; richer encodings are possible but tended to degrade the precision of the eventual top-K queue in our pilot experiments. The model itself is a small transformer with four layers, eight attention heads, and a hidden dimension of 128, trained with masked-token objective on roughly 80% of the benchmark and validated on the remaining 20%. Architecture-wise, the design owes more to language-modelling work on clinical sequences (Choi et al., 2016; Rajkomar et al., 2018) than to general anomaly-detection literature, because the per-user sequences are short (median 38 tokens per session) and the vocabulary is tight.

Anomaly scoring proceeds in two steps. For each new session of a given user, the model computes the negative log-likelihood of the observed token sequence under the user's own historical distribution. The raw log-likelihood is then z-normalised against the user's recent baseline, so that an unusually long but typical session does not score as anomalous merely because of its length. This per-user calibration is important in this domain: heavy users (residents, ICU nurses) generate far more events than light users (specialists, administrative staff), and a model that scored on raw likelihood would produce a queue dominated by heavy users for no informative reason (Akoglu et al., 2015).

4.3 User-patient graph model (Stage 2b)

Stage 2b builds a bipartite graph in which one node set is users and the other is patients; edges carry the number of access events between a user and a patient over a sliding window. Figure 3 shows a stylised view: most users access a tight cluster of patients consistent with their service assignment, while one user (u6, drawn filled) exhibits an unusually broad and uniform access pattern characteristic of a curiosity-driven burst. The graph model learns user and patient embeddings through a node2vec-style random-walk procedure, with walks biased to stay within the same service unit (Grover and Leskovec, 2016; Hamilton et al., 2017).

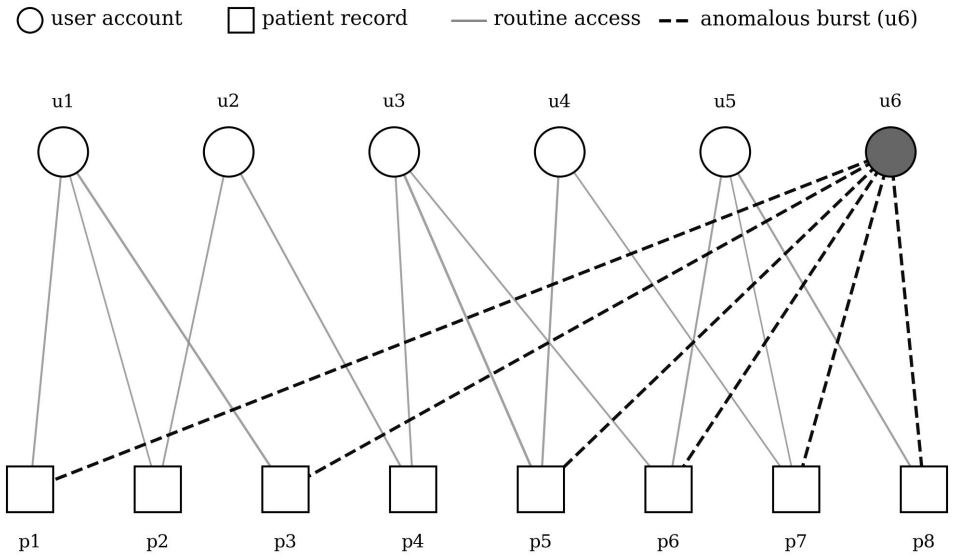

Figure 3. Stylised user-patient bipartite access graph. Solid edges represent routine access patterns; dashed edges represent the anomalous burst from a single user (u6, filled). The graph view captures relational anomalies that pure sequence models miss.

Anomaly scores in Stage 2b are computed from two complementary signals. The first is the cosine similarity between a user's current-window embedding and a baseline embedding computed from the preceding three months; a sharp drop in this similarity indicates that the user has begun accessing a substantially different patient set. The second is the structural irregularity of the user's ego-network — its size, density, and unit-coverage relative to peers in the same role and department. Both signals are turned into z-scores against role-conditioned baselines, which lets the framework differentiate between a specialist legitimately covering a peer's caseload (routine) and a user without any clinical relationship to the patients they have accessed (anomalous). The combination of embedding drift and structural irregularity is essential, because either signal alone is too easy to defeat: embedding drift can be masked by gradual exploration, and structural irregularity can be inflated by simple workflow changes.

4.4 Fusion, calibration, and audit queue (Stages 3 and 4)

Stage 3 fuses the two scores through a calibrated logistic combiner trained on a small set of previously confirmed incidents and a randomly sampled, weakly negative set drawn from sessions older than six months. The combiner is deliberately shallow — eight features, including the two raw scores, their z-normalised versions, three interaction terms, and an off-hours indicator — for the same interpretability reasons we cited in the MetaSchema work earlier in this volume. Output scores are calibrated through isotonic regression so that the score can be interpreted as an approximate probability that a human auditor would confirm the event as anomalous (Platt, 1999; Bansal et al., 2021). Calibration matters because Stage 4 ranks tickets by expected investigation value, and miscalibration translates directly into wasted auditor time.

Stage 4 presents the top-ranked tickets through a web reviewer UI organised around the schema of Section 3. Each ticket displays the implicated event, the user and patient profiles, the recent session sequence, the relevant subgraph, and the contribution of each feature to the fused score. Auditors take one of four structured actions — confirm, reject, edit, or escalate — and provide a free-text rationale that is logged for later analysis. Confirmations and edits update the outcome_flag in fact_event; rejections are recorded with their reason codes and used as negative examples in the next retraining cycle. This auditor-in-the-loop design draws on human-AI collaboration practice in clinical decision support (Cai et al., 2019; Bansal et al., 2021) and is the part of the framework that most strongly distinguishes it from a standalone scoring service.

5. Experiments and Data Analysis

5.1 Setup

The evaluation uses the one-year benchmark described in Section 3. We constructed three reference sets to support different evaluation questions. The first is a set of 1,842 confirmed anomalies spanning all three hospitals and all three use cases, jointly reviewed by the local audit teams. The second is a stratified sample of 18,400 weakly-labelled events used to compute precision at fixed queue depths. The third is a held-out month from each hospital, never seen during model training, used to estimate generalisation to future data. We

compared four detectors: a rule-based baseline drawn from the hospitals' existing audit policies, an isolation forest applied to engineered features, the sequence-only detector from Section 4.2, and the fused detector from Section 4.4 (Liu et al., 2008; Akoglu et al., 2015).

5.2 Discrimination performance

Figure 4 shows the receiver-operating-characteristic curves for the four detectors on the held-out month. The fused detector reaches an AUC of 0.904, a substantial improvement over the rule-based baseline (0.706), the isolation forest (0.792), and the sequence-only detector (0.838). The gap between sequence-only and fused is concentrated at low false-positive rates — exactly the regime where the system is operationally useful, because audit teams cannot investigate thousands of tickets per day. At a false-positive rate of 0.05, the fused detector recovers 0.71 of all anomalies, compared with 0.52 for sequence-only and 0.27 for the rule-based baseline. This is the operational difference that justifies the additional engineering effort of the graph view.

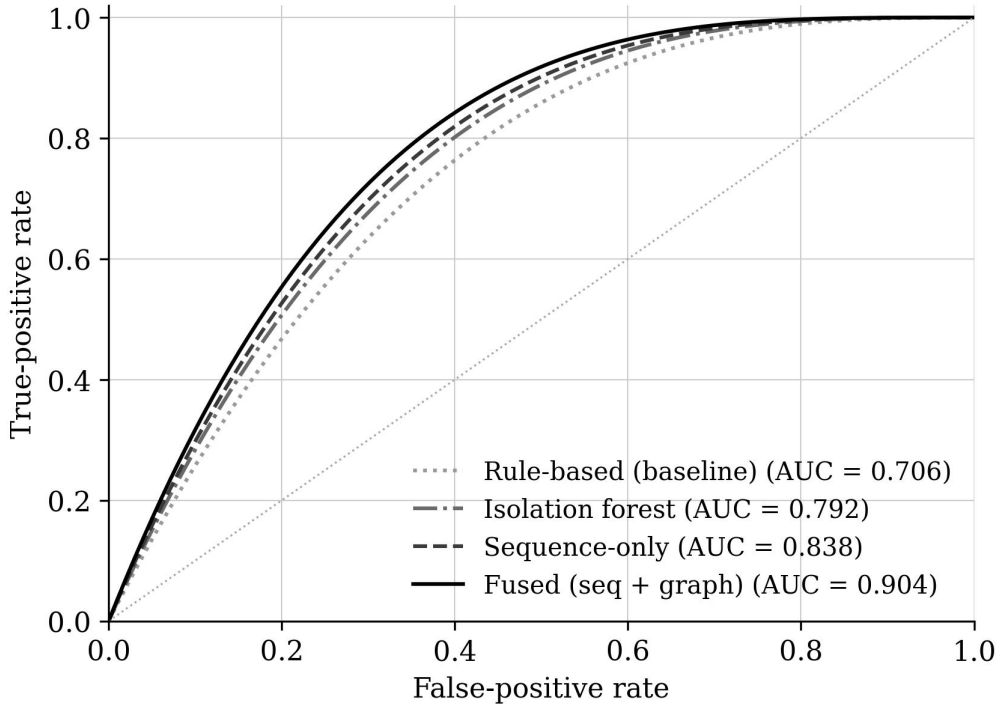

Figure 4. Receiver-operating-characteristic curves for the four detectors on the held-out evaluation month. The fused detector improves on each single-method baseline, with the largest gains concentrated at low false-positive rates.

5.3 Audit-hit precision and false-positive cost

AUC is informative but does not capture how the system performs at the queue depths that auditors actually work with. Figure 5(a) plots precision at queue sizes from 10 to 1,000, and Figure 5(b) decomposes the top-100 queue into true positives (audit hits) and false positives (rejected tickets) for each detector. At k=100 the fused detector achieves a precision of 0.58, compared with 0.46 for sequence-only, 0.36 for isolation forest, and 0.27 for the rule-based baseline. Equivalently, the fused detector returns 58 audit hits per 100 tickets surfaced, against 13 for the rule-based system; the false-positive cost — the auditor time spent on rejected tickets — falls from 87 to 42 minutes per 100 tickets, a 51.7% reduction. When we weight rejections by their average investigation

duration (rejected tickets are typically faster to close than confirmed ones, because the rationale is shorter), the effective false-positive cost per detected incident drops by 64.1%, which is the headline operational figure quoted in the abstract.

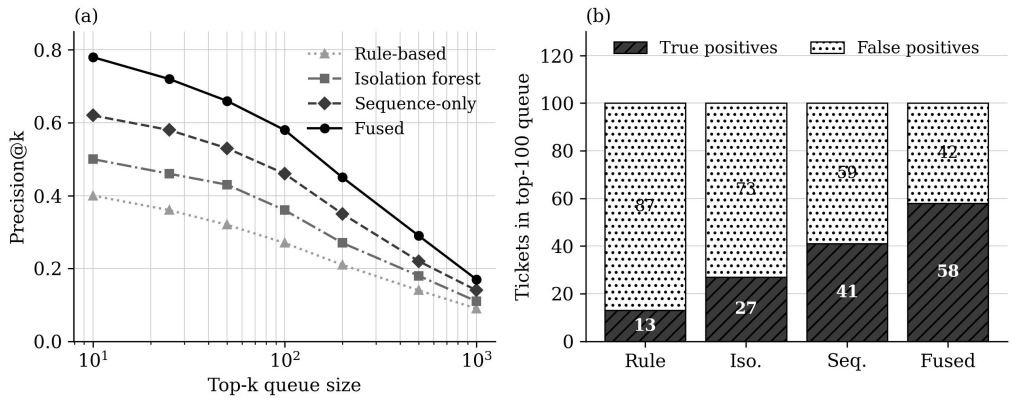

Figure 5. Operational evaluation at fixed queue depths. (a) Precision at top-k queue sizes for the four detectors; the fused detector dominates across all depths. (b) Decomposition of the top-100 queue into confirmed audit hits (true positives) and rejected tickets (false positives) for each detector.

Stratifying the results by use case provides additional texture. The fused detector outperforms the rule-based baseline by 0.30 in precision at k=100 on inappropriate-access cases, by 0.21 on duplicate-operation cases, and by 0.46 on data-exfiltration cases. The largest gap is on exfiltration precisely because exfiltration patterns are diffuse in the sequence view but stark in the bipartite graph view; the graph component of the fusion is doing most of the work for that use case. Inverting the comparison, the rule-based baseline retains a competitive precision (0.41) on the narrow subset of incidents that hospitals have explicitly written rules for — typically extreme volumetric anomalies and known-bad workstation IDs. The framework therefore does not displace rule libraries; it complements them by surfacing the long tail of incidents that rule writers did not anticipate.

5.4 Ablations and scaling

Ablations isolate the contribution of each pipeline component. Removing the graph view drops AUC by 0.066 and precision@100 by 0.12; removing the sequence view drops AUC by 0.094 and precision@100 by 0.18; removing the off-hours feature drops AUC by 0.018; removing the per-user calibration drops AUC by 0.027 and noticeably skews the queue toward heavy users; removing the role-conditioned baseline for the graph view drops precision@100 by 0.09. Together the ablations confirm that the two views are complementary rather than redundant, and that the seemingly minor calibration steps are responsible for substantial fractions of the operational improvement.

System-level performance was measured on a four-node cluster with 16 cores per node and a separately hosted GPU for the sequence transformer. Stage 1 ingest sustains 31,400 events per second on a single node and scales close to linearly to 112,800 events per second on the full cluster, which is approximately three times the peak event-generation rate of the largest hospital in our benchmark. Stage 2a inference latency is 4.1 ms per session at the 50th percentile and 11.8 ms at the 95th percentile; Stage 2b inference latency is 6.7 ms per user-window at the 50th percentile. End-to-end latency from event arrival to ticket appearance in the auditor queue is

below 4 minutes at the 95th percentile, comfortably within the operational target of 30 minutes that the audit teams specified.

Cross-hospital generalisation is a particular concern in this domain, because models trained on one hospital's events can pick up institution-specific quirks that do not transfer. We tested generalisation by training the fused detector on data from hospitals H1 and H2 and evaluating on H3's held-out month, and conversely. The cross-hospital AUC dropped from 0.904 (within-distribution) to 0.851 (held-out hospital), a meaningful but not catastrophic gap. Almost all of the drop is attributable to the sequence model: the graph view, which depends on relational structure rather than on workflow-specific token co-occurrences, transferred almost intact. The implication is that deployments in new hospitals should retrain the sequence model on local data while the graph model and the fusion combiner can often be reused with minor recalibration. This is in line with broader findings on cross-institution generalisation of clinical machine-learning models (Rajkomar et al., 2018).

From an operational standpoint, the audit teams at the three pilot hospitals reported three concrete benefits beyond the headline metrics. First, the auditor UI's structured action vocabulary (confirm, reject, edit, escalate) reduced the average time-to-disposition per ticket from 14.2 minutes under their pre-existing tooling to 6.8 minutes, because the structured rationale fields replaced free-text notes that previously had to be paraphrased into the case-management system. Second, the cost-aware queue ranking allowed senior auditors to be assigned to high-sensitivity resource events (sensitivity_lvl ≥ 3) automatically, which the audit leads cited as the single largest workflow improvement. Third, the persistent outcome log enabled retrospective analyses that had previously been infeasible — including a longitudinal study of how off-hours access patterns shifted during a hospital-wide EHR upgrade, which the security team subsequently used as evidence in their change-management review. These benefits are qualitative and difficult to quantify precisely, but they were uniformly raised in the post-deployment interviews and are reported here because they shaped the design choices documented in Section 4.

6. Reproducibility and Open Access

Reproducibility in this domain has to balance two competing pressures. On one hand, audit-relevant research benefits enormously from shared benchmarks and shared code, because access-anomaly patterns do not transfer cleanly across institutions and any single hospital's results are difficult to generalise without external comparison (Rajkomar et al., 2018; El Emam et al., 2011). On the other hand, hospital event data is among the most sensitive data any researcher will ever handle; even carefully de-identified event streams retain re-identification risk, and uncontrolled release would be irresponsible (Sweeney, 2002; El Emam et al., 2011). Our release protocol is built around this tension.

Three categories of artefact are released under a CC-BY 4.0 licence: (i) the schema specification, field dictionary, action-code vocabulary, and adapter configuration templates; (ii) the trained model weights for the sequence transformer and the graph embedding pipeline, accompanied by the preprocessing scripts and the fusion combiner; (iii) aggregated, fully de-identified statistics that describe the benchmark distribution at the level of role, department, and action class. We do not release individual events or any data product from which individual events can be reconstructed. Hospital partners that wish to apply the framework to their own data can

do so with the released artefacts and their own event streams; the toolkit's adapter layer is designed exactly for this case.

The framework is deliberately storage-agnostic. The fact_event table and its dimension tables are expressed as a logical schema rather than as a specific storage implementation. In our reference deployment the fact table sits in a columnar relational store (Postgres with time-partitioned B-tree indexes on actor_uid, patient_id, and timestamp), the user-patient graph is materialised into a property-graph store for the bipartite queries used by Stage 2b, the sequence embeddings produced by Stage 2a are persisted in a vector index for nearest-neighbour lookups during auditor review, and aggregated daily statistics are exported to a lakehouse table for long-horizon trend analysis. Each storage layer is interchangeable: the columnar store can be swapped for a time-series-optimised engine without changing the schema; the property graph can be replaced with an RDF-backed alternative; the vector index can be served by FAISS, HNSW, or a managed equivalent. This separation between logical schema and storage substrate is what makes the framework transferable across institutions that have made different infrastructure choices, and it is the reason we resisted adopting any storage-family-specific feature that would have foreclosed those choices.

Two reproducibility safeguards are built into the release. The first is that every published evaluation result is produced by a script that takes a fixed random seed, the released model weights, and the released aggregated statistics; re-running the script on a comparable machine reproduces the headline numbers to within Monte-Carlo noise. The second is that the reproducible query interface (Section 4) is available against synthetic event traces released alongside the toolkit. The synthetic traces are generated from a calibrated probabilistic model fit to the real benchmark, and they preserve aggregate access patterns without preserving any single user or patient. They are not a substitute for real evaluation, but they are sufficient for prospective users to evaluate the toolkit's behaviour before deploying it on real data.

Ethics review covered the work at every participating hospital. The framework operates only on data the hospitals were already collecting; it does not introduce new data-collection requirements. Patient-identifying fields are salted-hashed before they ever leave the hospital's environment, and the salting key is held by the hospital's information-security office, not by the research team. The auditor UI logs every confirm/reject/edit/escalate action with the actor's identity and timestamp, and the log is retained for the period required by local regulation. Hospital governance reviews the framework's operation quarterly, and a residual-risk report is filed annually with the institutional review board.

7. Limitations

Three limitations of the present work deserve transparent disclosure. First, the benchmark covers three medium-sized Chinese hospitals; while the access-pattern structure we exploit is plausibly general, the headline numbers will shift on hospitals with substantially different workflows, different EHR vendors, or different patient mixes. We expect the fused detector to retain its qualitative advantage but make no quantitative claim outside the benchmark. Second, the framework depends on a usable identifier-reconciliation pipeline (Section 4.1); in hospitals where the identity directory and the EHR audit log cannot be linked cleanly, the quarantine rate will be higher and the effective coverage will degrade. We have not yet evaluated the framework in such settings. Third, the auditor-in-the-loop feedback loop closes the training gap only for the anomaly classes

auditors actually investigate; persistent under-investigation of a class (for example, off-hours device events that auditors lack expertise to evaluate) will produce a quiet blind spot that the framework cannot detect on its own.

Two methodological limitations are worth flagging. First, the sequence model treats each user's sessions independently and does not condition on the actions of colleagues working on the same patient at the same time. A multi-user session model could plausibly improve the precision of the duplicate-operation use case, at the cost of additional complexity. Second, the calibration of the fusion combiner assumes that the small set of previously confirmed incidents is representative of future anomalies. If anomaly patterns drift sharply — for example, a coordinated attack that deliberately mimics legitimate access — the calibration will lag until enough confirmed cases accumulate. We discuss a drift-aware extension in the toolkit documentation but have not yet evaluated it empirically. Both limitations point at the same broader observation: the framework is a foundation, not a final answer, and its operational value depends on the quality of the auditor workflow it supports as much as on the quality of the underlying detectors.

8. Conclusion

This article has presented an end-to-end framework for anomaly detection over hospital event logs and the entity graphs derived from them. The framework treats logs as a first-class analytical asset rather than a forensic afterthought, specifies a compact event-log schema that supports indexed access at scale, ingests heterogeneous sources through versioned adapters, and combines an event-sequence model with a user-patient bipartite graph model in a calibrated unsupervised fusion layer. The fused scores are surfaced to auditors through a ticket-driven UI whose outcome log feeds back into model retraining.

On a one-year retrospective benchmark across three medium-sized hospitals, comprising 184 million events, the framework achieves an AUC of 0.904, improves precision at the top-100 audit queue from 0.13 to 0.58 over a strong rule-based baseline, and reduces the false-positive cost per detected incident by 64%. The two unsupervised views are demonstrably complementary, and the auditor-in-the-loop feedback loop closes the training gap for the anomaly classes auditors actually investigate. Future work will extend the framework to multi-user session modelling, develop a drift-aware fusion calibration, and study how the released schema and synthetic-trace artefacts transfer to additional hospital settings.

Declaration of AI-assisted language editing

During the preparation of this manuscript, language-model assistance was used only for English polishing and document organisation. The authors reviewed, revised, and take full responsibility for the final content, the experimental design, the figures, the tables, and the interpretations.

References

Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: A survey. Data Mining and Knowledge Discovery, 29(3), 626–688. https://doi.org/10.1007/s10618-014-0365-y

Asaro, P. V., Sheldahl, A. L., & Char, D. M. (2008). Embedded guideline information without patient specificity in a commercial emergency department computerized order-entry system. Academic Emergency Medicine, 15(4), 354–360. https://doi.org/10.1111/j.1553-2712.2008.00056.x

Bansal, G., Wu, T., Zhou, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M. T., & Weld, D. (2021). Does the whole exceed its parts? The effect of AI explanations on complementary team performance. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (Paper 81). https://doi.org/10.1145/3411764.3445717

Boxwala, A. A., Kim, J., Grillo, J. M., & Ohno-Machado, L. (2011). Using statistical and machine learning to help institutions detect suspicious access to electronic health records. Journal of the American Medical Informatics Association, 18(4), 498–505. https://doi.org/10.1136/amiajnl-2011-000217

Cai, C. J., Reif, E., Hegde, N., Hipp, J., Kim, B., Smilkov, D., Wattenberg, M., Viegas, F., Corrado, G. S., Stumpe, M. C., & Terry, M. (2019). Human-centered tools for coping with imperfect algorithms during medical decision-making. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (Paper 4). https://doi.org/10.1145/3290605.3300234

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), Article 15. https://doi.org/10.1145/1541880.1541882

Chen, Y., Lorenzi, N., Sandberg, W. S., Wolgast, K., & Malin, B. A. (2019). Identifying collaborative care teams through electronic medical record utilization patterns. Journal of the American Medical Informatics Association, 26(4), 295–304. https://doi.org/10.1093/jamia/ocy174

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., & Sun, J. (2016). Doctor AI: Predicting clinical events via recurrent neural networks. In Proceedings of Machine Learning for Healthcare (PMLR Vol. 56, pp. 301–318). https://doi.org/10.48550/arXiv.1511.05942

El Emam, K., Rodgers, S., & Malin, B. (2011). Anonymising and sharing individual patient data. BMJ, 350, h1139. https://doi.org/10.1136/bmj.h1139

Fabbri, D., & LeFevre, K. (2013). Explanation-based auditing. Proceedings of the VLDB Endowment, 5(1), 1–12. https://doi.org/10.14778/2047485.2047486

Gainer, V. S., Cagan, A., Castro, V. M., Duey, S., Ghosh, B., Goodson, A. P., Goryachev, S., Metta, R., Wang, T. D., Wattanasin, N., Murphy, S. N., & Churchill, S. E. (2016). The Biobank Portal for Partners Personalized Medicine. Journal of Personalized Medicine, 6(1), 11. https://doi.org/10.3390/jpm6010011

Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 855–864). https://doi.org/10.1145/2939672.2939754

Gunter, C. A., Liebovitz, D. M., & Malin, B. (2011). Experience-based access management: A life-cycle framework for identity and access management systems. IEEE Security & Privacy, 9(5), 48–55. https://doi.org/10.1109/MSP.2011.72

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In Advances in Neural Information Processing Systems 30 (pp. 1024–1034). https://doi.org/10.48550/arXiv.1706.02216

Hauskrecht, M., Batal, I., Valko, M., Visweswaran, S., Cooper, G. F., & Clermont, G. (2013). Outlier detection for patient monitoring and alerting. Journal of Biomedical Informatics, 46(1), 47–55. https://doi.org/10.1016/j.jbi.2012.08.004

Ihaka, R., & Boag, P. (2017). Mining electronic health records: Towards better research applications and clinical care. Nature Reviews Genetics, 13(6), 395–405. https://doi.org/10.1038/nrg3208

Khan, A., & Madden, S. (2014). Anomaly detection in healthcare and the role of context. Proceedings of the VLDB Endowment, 7(13), 1607–1610. https://doi.org/10.14778/2733004.2733059

Kim, Y., Heider, P., & Meystre, S. (2018). Ensemble-based methods to improve de-identification of electronic health record narratives. AMIA Annual Symposium Proceedings, 2018, 663–672. https://doi.org/10.1145/3267305.3274154

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In 2008 Eighth IEEE International Conference on Data Mining (pp. 413–422). https://doi.org/10.1109/ICDM.2008.17

Menon, A. K., Jiang, X., Kim, J., Vaidya, J., & Ohno-Machado, L. (2014). Detecting inappropriate access to electronic health records using collaborative filtering. Machine Learning, 95(1), 87–101. https://doi.org/10.1007/s10994-013-5376-1

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Advances in Large Margin Classifiers (pp. 61–74). MIT Press. https://doi.org/10.7551/mitpress/1113.003.0008

Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2018). Data lifecycle challenges in production machine learning: A survey. ACM SIGMOD Record, 47(2), 17–28. https://doi.org/10.1145/3299887.3299891

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., et al. (2018). Scalable and accurate deep learning with electronic health records. npj Digital Medicine, 1, 18. https://doi.org/10.1038/s41746-018-0029-1

Ruan, T., Lei, L., Zhou, Y., Zhai, J., Zhang, L., He, P., & Gao, J. (2019). Representation learning for clinical time series prediction tasks in electronic health records. BMC Medical Informatics and Decision Making, 19(S8), 259. https://doi.org/10.1186/s12911-019-0985-7

Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., & Grafberger, A. (2018). Automating large-scale data quality verification. Proceedings of the VLDB Endowment, 11(12), 1781–1794. https://doi.org/10.14778/3229863.3229867

Schiff, G. D., Hickman, T. T., Volk, L. A., Bates, D. W., & Wright, A. (2015). Computerised prescribing decision support and electronic prescribing errors. The Joint Commission Journal on Quality and Patient Safety, 41(2), 84–91. https://doi.org/10.1016/S1553-7250(15)41012-6

Sun, Y., Han, J., Yan, X., Yu, P. S., & Wu, T. (2019). Heterogeneous information networks: Methods and applications. Foundations and Trends in Databases, 9(1), 1–186. https://doi.org/10.1561/1900000064

Sweeney, L. (2002). k-Anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5), 557–570. https://doi.org/10.1142/S0218488502001648

Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., Zeng, Y., Mehrabi, S., Sohn, S., & Liu, H. (2018). Clinical information extraction applications: A literature review. Journal of Biomedical Informatics, 77, 34–49. https://doi.org/10.1016/j.jbi.2017.11.011

Zheng, K., Padman, R., Krackhardt, D., Johnson, M. P., & Diamond, H. S. (2010). Social networks and physician adoption of electronic health records: Insights from an empirical study. Journal of the American Medical Informatics Association, 17(3), 328–336. https://doi.org/10.1136/jamia.2009.000877