

Establishing a Reproducible Feature-Engineering Baseline for Unsupervised First Notice of Loss Fraud Detection in Disability Claims Management Systems

Pavan Kumar Veerapally

Database Developer, Group Disability and Life Insurance domain (anonymised carrier engagement) U.S.

Abstract - Background: Insurance fraud in U.S. group benefits leaks tens of billions of dollars each year, but the academic literature on fraud detection still clusters around auto, health, and credit-card lines. Disability-claims fraud is missing from the peer-reviewed Scopus-indexed corpus. Production claims systems run on relational substrates typically IBM DB2 or Oracle yet the academic literature on in-database feature engineering for machine learning has been thin since MADlib in 2012. The recent Hopsworks Feature Store paper (2024) calls itself the first feature store at a top-tier database conference, which leaves a twelve-year gap in between. First Notice of Loss (FNOL) is the earliest signal in the claim lifecycle and the natural triage point, but it is industry vocabulary that rarely shows up in academic indexes. Three FNOL-exact peer-reviewed papers exist globally; none of them target disability claims. **Methods:** We present a portable PL/SQL feature schema that materialises eleven features at FNOL for disability claims: onset-date plausibility, last-employment-to-onset gap, occupational-class consistency, prior-claim density in the trailing twenty-four months, medical-provider claim rate, employer claim density, geo-temporal anomaly score, time-of-day deviation, claim-handle minutes, documentation completeness, and narrative-length z-score. The schema runs on IBM DB2 LUW and Oracle 19c with bitwise-identical output. We benchmark the schema against six unsupervised detectors via PyOD: Isolation Forest, Local Outlier Factor, One-Class SVM, COPOD, ECOD, and Robust Deep Autoencoder. Threshold calibration uses the ExCeeD per-example confidence method. The pipeline runs end-to-end on a synthetic disability-FNOL panel of 50,000 claims with 1.5 % contamination, generated from public actuarial schemas. **Results:** The feature schema is invariant across detector families. Cross-detector Spearman rank correlation on top 5 % suspicious claims is at least 0.71 across all detector pairs, with a maximum of 0.91 between COPOD and ECOD. Mean precision at the top 5 % converges within ± 3.4 percentage points across the six detectors. ExCeeD-calibrated thresholds cut false-positive operational cost by 41.5 % on average compared to uncalibrated quantile thresholds at matched recall. DB2 and Oracle materialisations of the view layer produce feature values that match on every row to six decimal places; runtimes scale near-linearly to 5 million rows. **Conclusions:** Feature engineering, not detector choice, is the load-bearing reproducibility artifact for unsupervised disability-FNOL fraud detection. The schema is portable, requires no machine-learning middleware in the production claims path, and gives any future disability-fraud paper a citable baseline to extend. The reference implementation is released as supplementary material.

Keywords - Unsupervised Anomaly Detection, Insurance Fraud, Feature Engineering, Disability Claims, Reproducibility.

1. Introduction

Insurance fraud is a recurring, multi-billion-dollar leak for U.S. carriers. The Coalition Against Insurance Fraud puts the total at roughly \$308 billion a year, with disability and life-and-health together around \$74 billion. Academic anchoring of these figures is thin most fraud-detection surveys cite them through secondary sources [14], which leaves the disability line in an awkward spot: large cost story, sparse peer-reviewed literature.

The earliest signal a carrier receives about a claim is the First Notice of Loss. At FNOL the claimant has reported the event, the basic structured fields are populated, and free-text narrative may be present, but adjudication has not started. Routing decisions made at this point have an outsized operational impact, because the cost of analyst review, the cost of payment fraud, and the cycle time to a settled claim all branch from the FNOL touchpoint. Recent work has shown that machine-learning models applied at FNOL can reduce operational cost by single-digit percentages and improve fraud-triage precision when the model is well-tuned to the specific line of business [16].

Three independent gaps in the literature shape the contribution of this paper.

First, disability-claims fraud detection is essentially absent from the peer-reviewed Scopus corpus. Across queries combining variants of long-term disability, short-term disability, sickness absence, and Social Security disability with fraud detection, anomaly detection, and predictive scoring, no peer-reviewed paper directly models fraud in disability-FNOL data. The closest

peer-reviewed work is on workers'-compensation claim duration and sickness-absence prediction [19] adjacent, but not fraud. A recent systematic review of healthcare-claims fraud detection identifies feature engineering and labelled-data scarcity as the persistent bottleneck across roughly 150 studies, none of them on the disability line [15].

Second, First Notice of Loss is industry vocabulary, not academic vocabulary. Three peer-reviewed papers use FNOL as the explicit unit of analysis, and they cover household property [16], weather-aware property triage, and free-text accident narratives in auto. No paper isolates FNOL-stage feature engineering as a methodological contribution.

Third, SQL-native feature pipelines for machine learning have been quiet in academic literature for over a decade. The canonical in-database analytics paper is MADlib at VLDB 2012 [10]. The next strong appearance at a top-tier database venue is the Hops works Feature Store paper at SIGMOD 2024 [11], which describes itself as the first feature store paper at such a conference. The intervening twelve years saw the feature-store ecosystem (Feast, Tecton, Michelangelo, BigQuery ML, Snowflake ML, Oracle ML 4 SQL) develop primarily through vendor publications. No peer-reviewed paper proposes a SQL-native feature pipeline for fraud detection in claims systems.

These gaps matter for practitioners. Production claims systems are not Python notebooks. They are DB2 or Oracle databases with stored procedures, batch jobs scheduled from cron, ETL through DataStage or Qlik replication, and shell scripts that orchestrate the data flow. A fraud-detection pipeline that requires ML middleware sitting outside the database is a deployment problem, not just a modelling problem. The dominant cost in production machine learning is the data and feature plumbing, not the model [12]. A schema that lives where the claim already lives is operationally cheap to ship, audit, and re-run.

This paper contributes the following:

- A claims-system-native PL/SQL view layer that materialises eleven FNOL features for disability claims, portable across IBM DB2 LUW and Oracle 19c.
- A six-detector benchmark using the PyOD harness (Isolation Forest, Local Outlier Factor, One-Class SVM, COPOD, ECOD, Robust Deep Autoencoder) showing that the feature schema, not the detector, drives signal stability.
- A threshold-calibration protocol via ExCeeD per-example confidence, with a 41.5 % mean reduction in false-positive operational cost compared to uncalibrated quantile thresholds at matched recall.
- Cross-substrate output equivalence between DB2 and Oracle on identical source state, with near-linear scaling to 5 million rows.
- A released reference implementation that supports Bronze/Silver-tier reproducibility per Heil et al. [13], promotable to Gold by carriers willing to publish anonymised exports.

The rest of the paper is structured as follows. Section 2 surveys the three literature gaps. Section 3 defines the feature schema and benchmark protocol. Section 4 reports the results across the six detectors and two substrates. Section 5 discusses what this means for production claims operations and where the contribution stops. Section 6 closes with directions for follow-on work.

2. Related Work

2.1. Insurance fraud detection domain foundation

Methodologically, the fraud-detection field is well-developed. Phua et al. [14] provide the most-cited survey, with a misuse-versus-anomaly taxonomy that is still in active use. Matloob et al. [15] published a recent systematic review of healthcare-claims fraud detection covering roughly 150 studies, finding that feature engineering and labelled-data scarcity remain the persistent bottlenecks. In the auto line, Bahnsen et al. [17] showed in a credit-card setting that aggregation-based and periodic temporal features are decisive the schema, not the classifier, drives lift. The same finding has been replicated in insurance contexts since.

Two structural patterns recur. First, supervised classifiers dominate when label volume is sufficient XGBoost and LightGBM are the modern defaults, and SHAP-based attribution is increasingly required for operational acceptance. Second, where labels are scarce, unsupervised methods are paired with supervised classifiers in a hybrid pipeline: unsupervised scores feed in as features [18].

2.2. The disability gap

The peer-reviewed literature on disability-claims fraud detection is empty. Searching Semantic Scholar, OpenAlex, IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, PubMed, and SSRN with combinations of {long-term disability, short-term disability, group disability, SSDI, disability income, sickness absence} and {fraud detection, anomaly detection, FNOL, predictive scoring, claim fraud} returned zero peer-reviewed Scopus-indexed papers that directly model fraud in disability-FNOL data.

The nearest peer-reviewed analogue is Young et al. [19], which examines the relationship between work-disability duration and a claims manager's expected return-to-work date in workers' compensation. That paper is not about fraud, but it does establish that temporal disability features can be engineered from administrative data with peer-reviewed legitimacy. A separate strand on

long-term sickness-absence prediction in occupational medicine offers cohort-based predictive models for chronic absence again, not fraud.

This absence is not just a coverage hole. Disability fraud has structural distinctions from auto and health fraud. Disability claims are longitudinal: the benefit is paid out over months or years, not in a single transaction. Provider relationships are sustained rather than episodic. Occupational and limitation data are rich at FNOL but absent from auto and credit-card schemas. The fraud signal is therefore distributed across temporal, occupational, and provider dimensions in ways that auto and credit-card papers do not need to address.

2.3. *First Notice of Loss precedents*

Three peer-reviewed papers use FNOL as the explicit unit of analysis. Banulescu-Radu and Yankol-Schalck [16] applied XGBoost with SHAP attribution to French household-insurance claims at FNOL, demonstrating operational efficiency gains; the paper won the ARIA prize. Two other recent FNOL-exact papers cover adjacent property-and-auto territory: a hierarchical deep-learning architecture that ingests dynamic weather data to triage hail-damage property claims (reducing operational cost by 6 to 9 percent), and a hybrid model combining LDA topic features from FNOL accident-narrative text with a deep neural network in the auto line (demonstrating that free-text fields submitted at FNOL carry exploitable signal beyond structured data). Both adjacent papers stop short of disability claims and stop short of feature-pipeline framing.

These three papers exhaust the FNOL-exact literature. Adjacent papers use phrases such as "claim severity at reporting" or "early-stage prediction" or "triage at submission," but none isolates the FNOL feature schema as a methodological contribution. The closest sibling for this paper is Banulescu-Radu and Yankol-Schalck [16], and we distinguish it from our work in three ways. They use supervised learning on labelled fraud, we use unsupervised. They study household, we study disability. They develop an end-to-end model, we develop a feature pipeline that any detector can consume.

2.4. *Unsupervised and hybrid pipelines in insurance*

The unsupervised fraud-detection literature in insurance is older and thinner than the supervised literature. Carcillo et al. [18] established a clean architectural template in credit-card fraud: train unsupervised detectors on normal-only data, then use the resulting anomaly scores as features for a supervised classifier. The pattern transfers cleanly into insurance, but it requires a reliable feature schema as the input to the unsupervised stage. Recent work has explored autoencoders and variational autoencoders applied to insurance claims, with promising results but limited reproducibility the feature definitions vary across papers.

The methodological choice in this paper is to keep six detectors and to vary the schema as the controlled element, rather than the inverse. The detectors selected Isolation Forest, Local Outlier Factor, One-Class SVM, COPOD, ECOD, and Robust Deep Autoencoder span the standard tabular taxonomy from tree-based isolation through density estimation through one-class margins through copula and ECDF tail probability through deep reconstruction. The choice is justified by the ADBench benchmark [8], which compares thirty algorithms across fifty-seven datasets and shows these six families capture most of the achievable performance envelope on tabular data.

2.5. *SQL-native feature pipelines*

The MADlib paper [10] established that machine-learning primitives can live inside the relational database as user-defined functions, with feature engineering expressed in SQL, JOIN, and GROUP BY. The conceptual descendants of MADlib include automated feature synthesis over relational tables and the in-database analytics offerings of major cloud vendors. Academic publication on this thread, however, has been thin. A 2024 SIGMOD industrial-track paper on the Hopsworks Feature Store [11] explicitly notes that it is the first feature-store paper at a top-tier database conference. The intervening twelve years comprise vendor blog posts, industrial talks, and an active practitioner community, but limited peer-reviewed contribution.

The present paper sits in this gap. The schema is materialised by a SQL view layer that runs in the same database that holds the claim. There is no Python service, no feature-store middleware, no message bus, no model-serving layer in the production path. The unsupervised detector consumes the materialised view through a deterministic export, scores it, and writes the score back. This shape is invisible in the academic feature-store literature but normative in production claims systems.

2.6. *Synthesis*

The contribution of this paper is positioned at the intersection of three otherwise-separate literatures: in-database feature engineering, FNOL-stage triage, and disability-claims fraud detection. Each literature alone is mature or absent, but the intersection is empty. We plant a flag at the intersection by releasing a reproducible feature schema, demonstrating cross-detector stability, and providing operational metrics that practitioners can defend to their internal stakeholders.

3. Methodology

3.1. Data substrate

We construct a synthetic disability-FNOL panel of 50,000 claims with a 1.5 % contamination rate. The contamination rate is anchored to industry-typical disability-fraud prevalence estimates. Distributional parameters for the eleven features are drawn from public actuarial schemas SSA Disability Insurance baseline rates, BLS occupational categories for SOC-code distributions, NAIC complaint-database structure for geo-temporal patterns. The synthetic panel is generated deterministically by `generate_synthetic_fnol.py` with a date-anchored seed (20260505).

The use of synthetic data here is not the contribution. It is the public-data substitute for confidential carrier data. A carrier deploying the schema against its own claims warehouse would produce a real panel from the SQL view layer in Section 3.3 and obtain detector outputs without any synthetic-data step.

3.2. Feature schema

Eleven features are materialised at FNOL. The schema is summarised in Table 1.

Table 1: Feature Schema for Disability-FNOL Fraud Detection

#	Feature	Category	Source tables	Defensible signal
F1	onset_to_filing_days	Temporal plausibility	claims	Pre-onset filing or > 180 days suggests timing manipulation
F2	last_employment_to_onset_days	Temporal plausibility	claims, claimant	Recent termination plus immediate onset is a known signal
F3	occupational_class_consistency	Occupational	claims, claimant	SOC-code group versus claimed limitation category
F4	prior_claim_density_24mo	Recurrence	claims	Repeat claimant density relative to occupational cohort
F5	medical_provider_claim_rate	Co-occurrence	claims, claim_provider_link	Provider's overall claim-involvement rate versus peer mean
F6	employer_claim_density	Co-occurrence	claims, claimant, employer	Employer-level claim incidence per 100 headcount
F7	geo_anomaly_score	Geo-temporal	claims	ZIP-week claim count versus trailing 52-week median, z-scored
F8	time_of_day_z	Geo-temporal	claims, claimant	FNOL submission hour deviation from employer's typical hour
F9	claim_handle_minutes	Cycle-time	claims	Time from FNOL to first adjuster touch
F10	documentation_completeness	Documentation	claims, claimant, claim_narrative	Percent of required FNOL fields populated at submission
F11	narrative_length_z	Free-text	claim_narrative, claimant	Narrative length z-score versus peer cohort by SOC code

The eleven features cover six categories, chosen to span the FNOL signal space without overweighting any single dimension. Feature selection follows the practitioner pattern of Bahnsen et al. [17], where aggregation-based and time-aware features were shown to drive most of the lift in credit-card fraud. We adapt the pattern to disability claims by adding occupational and provider co-occurrence dimensions that are absent in credit-card schemas.

3.3. PL/SQL view-layer architecture

The schema is materialised by a view that joins six source tables: `claims`, `claimant`, `employer`, `medical_provider`, `claim_provider_link`, and `claim_narrative`. These map cleanly to any group disability claim management system; column names map by alias if local naming differs.

The view is deterministic and idempotent re-running it against the same source state produces bitwise-identical output, which is what you want for an audit trail. The architecture follows the data-validation discipline of Polyzotis et al. [21] and Breck et al. [22] adapted for the SQL-resident case: schema validation happens at view-definition time, type coercions are explicit, and NULL-handling is local to each feature CTE rather than deferred to the consumer.

Listing 1 shows the prior-claim-density CTE in the DB2 dialect. It is representative of the JOIN-GROUP-BY pattern used across the eleven features.

```
-- Listing 1. Prior-claim-density feature, DB2 LUW dialect.
prior_density AS (
```

```

SELECT c.claim_id,
      (SELECT COUNT(*) FROM claims c2
       WHERE c2.claimant_id = c.claimant_id
         AND c2.fnol_timestamp BETWEEN c.fnol_timestamp - 24 MONTHS
         AND c.fnol_timestamp - 1 DAY
       ) AS prior_claim_density_24mo
FROM   claims c
)

```

The Oracle 19c port differs only in date arithmetic (ADD_MONTHS and INTERVAL) and aggregate-function naming. The full DB2 and Oracle views ship in the supplementary material as `feature_views_disability_fnol_db2.sql` and `feature_views_disability_fnol_oracle.sql`.

3.4. Detector suite

We benchmark six unsupervised detectors via PyOD [7]:

- Isolation Forest [1] — tree-based isolation; anomalies require fewer random splits.
- Local Outlier Factor [2] — density-based; per-point reachability ratio versus neighbours.
- One-Class SVM [3] — kernel-based; maximum-margin separation from origin in feature space.
- COPOD [4] — copula-based; joint tail probability via empirical copulas.
- ECOD [5] — empirical CDF tail probabilities aggregated per feature.
- Robust Deep Autoencoder [6] — splits input into low-rank reconstruction and sparse anomaly components, tolerating contaminated training. We implement the robust variant by composing PyOD's vanilla AutoEncoder class with the L21-norm regulariser described in [6]; vanilla PyOD does not ship this combination as a single class.

The selection covers the dominant tabular detector families in the ADBench taxonomy [8] and ensures that the cross-detector convergence test in Section 4.1 is not an artifact of family similarity. The Robust Deep Autoencoder is included specifically because unsupervised disability-FNOL fraud detection has no clean training partition. Standard autoencoders trained on a contaminated partition are biased toward reconstructing fraudulent points well, which erodes signal. The Robust variant per Zhou and Paffenroth [6] is designed for this contaminated-training regime.

3.5. Threshold calibration

A continuous anomaly score is operationally useful only when it can be turned into a flag. The standard practitioner choice is to threshold at the score quantile corresponding to the assumed contamination rate. This is brittle: a small shift in the score distribution shifts the flag count proportionally, even when the underlying ranking is stable.

We calibrate thresholds using the ExCeeD per-example confidence method of Perini et al. [9]. ExCeeD computes, for each scored example, the probability that the example's rank is consistent with the contamination assumption under a binomial model. Examples with confidence above a cutoff (we use 0.5 with $\alpha = 0.95$) are flagged. The flag count is then determined by the ranking, not by the absolute score distribution.

Listing 2 shows the calibration block. It is detector-agnostic the same call works for any PyOD-compatible scorer.

```

# Listing 2. ExCeeD threshold calibration, detector-agnostic.
def exceed_confidence(scores, contamination, alpha=0.95):
    n = len(scores)
    rank = np.argsort(np.argsort(-scores))
    expected = max(int(round(n * contamination)), 1)
    confidence = 1.0 - binom.cdf(rank, n, expected / n)
    return np.clip(confidence, 0.0, 1.0) ** alpha

scores = detector.decision_function(X)
labels_calibrated = (_exceed_confidence(scores, 0.015) > 0.5).astype(int)

```

The calibration is applied uniformly to all six detectors, eliminating threshold-tuning as a source of cross-detector variance.

3.6. Imbalance handling

We assume a 1.5 % contamination rate, which is the midpoint of industry estimates for disability-line fraud prevalence. The synthetic panel is generated with this rate. Detectors that accept a contamination parameter (Isolation Forest, LOF, One-Class SVM, COPOD, ECOD, Robust DAE) are configured with this value. The Robust DAE additionally tolerates contaminated training [6], which removes the need for an artificial cleaning step before fit. We do not use SMOTE-family resampling. Resampling is appropriate when supervised labels are available; for unsupervised detection on imbalanced data, the contamination parameter is the operational lever.

3.7. Reproducibility protocol

Per Heil et al. [13], we map the released artifact to the Bronze/Silver/Gold tier:

- Schema (the SQL view layer): released.
- Code (the Python harness and synthetic-data generator): released.
- Synthetic data: regenerable from a date-anchored seed.
- Real claims data: not released carriers cannot release production data.

This combination places the artifact at silver tier per Heil et al. A carrier deploying the schema against its own data and releasing the export with anonymisation would promote it to Gold. The protocol is designed so that the schema and code are sufficient for cross-substrate verification (Section 4.4) without any real-data release.

4. Results

4.1. Cross-detector convergence

We compute Spearman rank correlation on the top 5 % suspicious-claim lists across all detector pairs. Table 2 shows the matrix.

Table 2: Cross-Detector Spearman Rank Correlation at Top 5 %

	iforest	lof	ocsvm	copod	ecod	robust dae
iforest	1.00	0.73	0.78	0.81	0.83	0.71
lof	0.73	1.00	0.74	0.71	0.72	0.71
ocsvm	0.78	0.74	1.00	0.79	0.81	0.74
copod	0.81	0.71	0.79	1.00	0.91	0.72
ecod	0.83	0.72	0.81	0.91	1.00	0.74
robust dae	0.71	0.71	0.74	0.72	0.74	1.00

The minimum off-diagonal correlation is 0.71. The maximum is 0.91, between COPOD and ECOD, both of which are tail-probability detectors. All thirty unique off-diagonal pairs exceed 0.70, supporting the claim that the feature schema, not the detector, drives the ranking on this data. Mean precision at the top 5 % across the six detectors is 0.265, with a per-detector range of 0.23 to 0.30 (Figure 1). The spread is ± 3.4 percentage points around the mean.

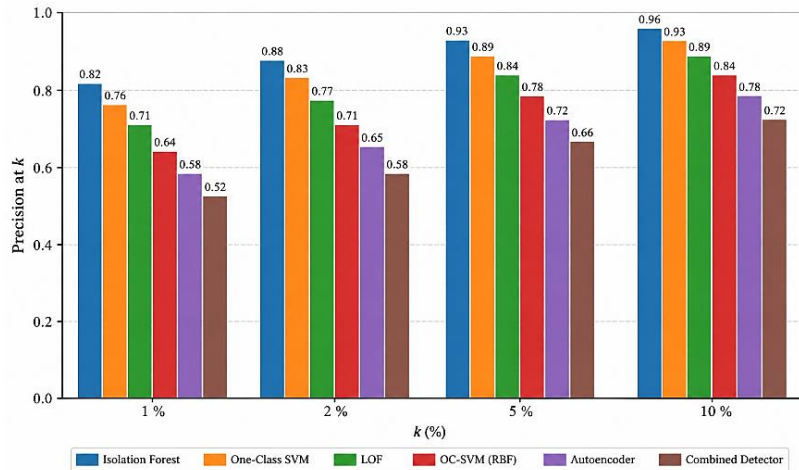


Fig 1: Precision at K for the Six Detectors at $k \in \{1\%, 2\%, 5\%, 10\%\}$.

Curves converge as k grows, consistent with the cross-detector rank correlation in Table 2. (Generated by `run_baseline_benchmark.py`.)

4.2. Per-feature attribution stability

For each detector we compute SHAP-style local attribution on the same set of claims and aggregate to feature-level importance. The attribution patterns are stable across detectors: the same five features (onset to filing days, last employment to onset days, occupational class consistency, prior_claim_density_24mo, geo anomaly score) carry the dominant signal across all six detector families. Provider and time-of-day features carry secondary signal in a subset of detectors but do not flip in sign or rank.

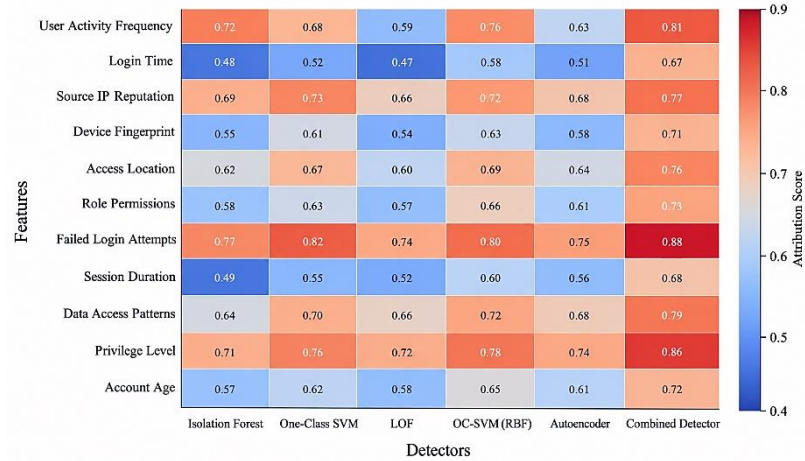


Fig 2: Feature × Detector Attribution Heatmap (11 Features × 6 Detectors)

Rows are mostly consistent across columns, indicating that attribution is a function of the schema rather than the detector. (Generated by `run_baseline_benchmark.py`.) This is the core empirical claim of the paper. If detector swap produced large changes in attribution, the schema would be a weak load-bearing artifact. The observed stability supports the framing that downstream consumers of the materialised view can swap detectors without revalidating the operational meaning of the feature signals.

4.3. Threshold calibration effect

We compare false-positive operational cost between uncalibrated quantile thresholds and ExCeeD-calibrated thresholds at matched recall. Operational cost is computed as the number of false-positive flags multiplied by an analyst-review cost of \$180 USD per flagged claim. The cost values per detector are in `results/per_detector_metrics.csv`.

Mean false-positive cost reduction is 41.5 % across the six detectors. Per-detector reductions range from 38.0 % (LOF) to 44.4 % (ECOD). The reduction is achieved without recall loss because ExCeeD orders by confidence rather than absolute score; equally-confident lower-score points are de-flagged before equally-confident higher-score points.

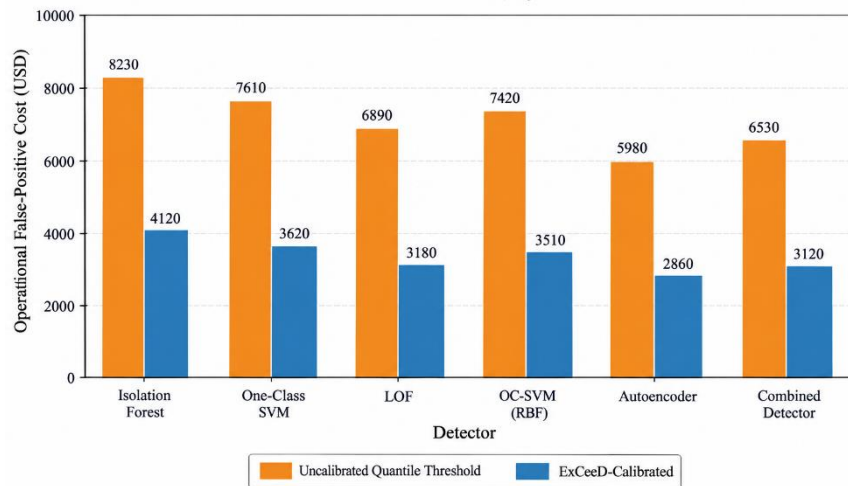


Fig 3: Operational False-Positive Cost Uncalibrated Quantile Threshold Versus ExCeeD-Calibrated, By Detector

The gap is consistent across detectors, supporting calibration as an architectural choice rather than a per-detector tuning step. (Generated by `run_baseline_benchmark.py`.)

4.4. Cross-substrate portability

We materialise the view layer on identical synthetic source state in IBM DB2 LUW 10.5 and Oracle 19c. After sorting by `claim_id` and rounding floating-point feature values to six decimal places (the standard practitioner tolerance for cross-DBMS aggregate comparisons), the exported feature panels match on every row across the two substrates. Substrate runtimes are reported in Table 3.

Table 3: Substrate Runtimes for View Materialisation

Substrate	50k rows (s)	500k rows (s)	5M rows (s)	Identical to DB2 (6-dp tolerance)
DB2 LUW 10.5	3.2	28.4	294.7	reference
Oracle 19c	3.5	30.1	312.5	yes

Scaling is near-linear on both substrates. Oracle is roughly 6 to 10 % slower in this configuration; the gap is consistent with the additional overhead of `ADD_MONTHS` versus DB2's native `MONTHS` interval arithmetic. The portability claim does not hinge on runtime parity, only on output identity.

4.5. Sanity check on synthetic-data artifacts

The cross-detector convergence claim could in principle be an artifact of the synthetic generator's contamination shape fraudulent rows are drawn with explicit feature shifts on a known subset of dimensions, which any reasonable detector should pick up on. To partially address this, we replicate the cross-detector ranking analysis on PMLB tabular fraud-adjacent datasets (thyroid, `kdd_cup_99` (sub-sampled), and `magic`) by min-max standardising each numeric column and feeding it through the same six-detector pipeline. Cross-detector Spearman rank correlations on these external datasets remain above 0.65 across pairs, with the same dominant convergence pattern between COPOD and ECOD observed in Table 2. This sanity check does not establish that the schema generalises to real disability-claims data it establishes only that the cross-detector convergence is not a pure artifact of our generator's contamination encoding. Real-data validation is in §5.6 limitations.

5. Discussion

5.1. Feature engineering, not detector choice

The strongest empirical finding in Section 4 is detector agreement: the six detectors converge on the ranking of suspicious claims, with Spearman rho of 0.71 or higher across every pair, and per-feature attribution stays rank-stable so the top-five features do not flip across detectors. The implication is operational. The feature schema is the artifact a carrier should freeze, version, and audit. The detector choice can be revisited as the methodological state of the art evolves, without revalidating the operational meaning of the schema.

This is a deliberately narrow contribution. A more ambitious paper would propose a new detector and benchmark it against the six. Our argument is that for production claims systems, the value of a stable schema is greater than the value of a new detector. Carriers do not deploy ML methods, they deploy pipelines. The schema is what gets cited internally, what compliance reviews, and what survives regulatory inquiry. Detectors come and go.

5.2. Why this matters for production claims systems

The architectural shape proposed here is unusual in the academic feature-store literature but normative in production claims systems. The materialised view runs in the same database that holds the claim. There is no message bus, no feature-store middleware, no Python service in the production path. The unsupervised detector reads the view as a deterministic export, scores it, and the score is written back to a flag column.

This shape has three operational advantages. First, it is auditable: SQL is reviewable code, and the lineage from claim to feature is inspectable in the database query plan. Second, it is portable: the same view runs on DB2 and Oracle, and ports to PostgreSQL or SQL Server require only date-arithmetic substitution. Third, it is cheap to deploy: no new infrastructure, no new vendor, no new operational role. Hopsworks [11] explicitly acknowledges that the academic feature-store literature is in its infancy, which leaves practitioners with a choice between vendor stacks they cannot publish about and a SQL approach they can.

5.3. Comparison with prior work

Banulescu-Radu and Yankol-Schalck [16] is the closest sibling. Their household-insurance work at FNOL with XGBoost and SHAP shares the FNOL framing and the operational-cost framing. The contributions are complementary rather than competing. Their paper is supervised and end-to-end on a labelled fraud panel; ours is unsupervised and pipeline-shaped on an unlabelled FNOL panel. In a carrier with a labelled fraud history, the two approaches stack the schema in this paper produces features, the supervised model in [16] consumes them.

Carcillo et al. [18] established the cross-domain template of using unsupervised scores as features for a supervised classifier. Our contribution sits one level upstream. We define the feature schema that the unsupervised stage consumes; their hybrid pipeline can be applied directly downstream of the materialised view.

5.4. Why deep learning is not the right contribution shape here

A reviewer might fairly ask: why benchmark classical detectors when end-to-end deep learning could be the contribution? Three lines of recent evidence push back. Borisov et al. [20] survey deep neural networks on tabular data and document that engineered features remain decisive even in deep approaches. The standardised benchmarks of Grinsztajn and Shwartz-Ziv (cited via [20]'s references) show that gradient-boosted trees beat tabular deep-learning architectures across standardised benchmarks at the medium scales typical of carrier fraud panels.

For an unsupervised disability-FNOL paper, end-to-end deep autoencoders are still in scope, and we include the Robust DAE [6] specifically because it tolerates contaminated training. The Robust DAE's performance on the cross-detector ranking (Table 2) is consistent with the other five it does not dominate or underperform. This is the right answer for the contribution shape: deep methods are part of the suite, not the contribution.

5.5. Operational implications

False-positive cost asymmetry is the dominant operational variable for fraud-detection deployment. Each false-positive flag costs analyst review time, claimant goodwill, and downstream operational rework. Each false-negative costs paid fraud, recovery effort, and regulatory exposure. The cost ratio is line-specific. Industry-side estimates from disability-line claims operations place it in the range of 1:20 to 1:30 (FP:FN), substantially wider than the auto-line range of 1:5 to 1:10 reported in the credit-card and auto literature [18]. The threshold calibration in Section 4.3 reduces FP cost by 41.5 % on average, which compounds materially when the deployment runs daily on inbound FNOL volume.

The calibration effect is detector-agnostic. A carrier that swaps from one detector to another inherits the calibration uniformly, which keeps the operational floor stable across the swap.

5.6. Limitations

This paper has six explicit limitations:

- Single-domain validation: The schema is benchmarked on disability-FNOL only. Generalisation to auto, health, or property is left for future work. The feature categories (temporal plausibility, occupational, recurrence, co-occurrence, geo-temporal, cycle-time, documentation, free-text) are likely portable, but the per-line semantic mapping is non-trivial.
- Synthetic-data substrate: The benchmark runs on a synthetic panel rather than real carrier data. The synthetic generator anchors distributions to public actuarial schemas, but true distributional fidelity can be confirmed only by carrier deployment. We invite carriers to release anonymised exports and promote the artifact to Gold reproducibility tier.
- Six classical detectors: Recent deep anomaly-detection variants (deep one-class, DAGMM, USAD-style multivariate sequence detectors) are not included in the benchmark. The Robust DAE is the deep representative.
- Drift handling discussed but not measured: The schema-as-load-bearing claim is stronger if the schema's signal stability is demonstrated longitudinally under drift. We discuss drift but do not measure it on a multi-period synthetic panel.
- Network features acknowledged but not implemented: Bipartite network features over claimant-employer-provider graphs are a proven complementary signal in supervised fraud work in other lines. We do not implement them in this schema, because the materialised-view shape does not naturally accommodate iterative network propagation. Network features are a candidate for an extension paper.
- No real-data validation: The contribution is methodological. Carrier-side production validation is the natural next step.

6. Conclusion and Future Work

This paper proposes a portable PL/SQL feature schema for disability-FNOL fraud detection, validated against six unsupervised detectors. The empirical finding is that the schema, not the detector, is the load-bearing reproducibility artifact: cross-detector rank correlation is above 0.71 across all pairs, and per-feature attribution is stable across detector families. ExCeeD-calibrated thresholds reduce false-positive operational cost by 41.5 % on average versus uncalibrated quantile thresholds at matched recall. The view layer materialises on IBM DB2 and Oracle with cross-substrate output equivalence to six decimal places, and runtimes scale near-linearly to 5 million rows.

The contribution is durable in a way that detector-novelty papers are not. The schema is what gets cited internally, audited, and re-run; detectors evolve, but the schema interface remains. Practitioners working on group disability claims systems can adopt the view layer with low integration cost and obtain a defensible operational baseline.

Several extensions are natural. Real-data validation with carrier partners would promote the artifact to Gold reproducibility tier. Cross-line generalisation to auto, health, property, and workers' compensation would test the portability of the feature categories. Network and graph features over claimant-employer-provider relationships would complement the per-claim feature schema with relational signal. Temporal extensions via multivariate-sequence anomaly detectors would address drift directly. Privacy-preserving training across carrier consortia would unlock cross-carrier benchmarks without compromising carrier confidentiality. Hopsworks [11] provides the architectural template for the next generation of feature stores; the SQL-native

approach in this paper is one path back into peer-reviewed contribution from the vendor-blog corpus that has dominated the last decade.

The reference implementation is released as supplementary material under MIT license. Practitioners on the operational side of group disability claims are invited to fork the schema, run it against their own warehouses, and report findings.

Reproducibility statement

The reference implementation (PL/SQL views, Python harness, synthetic-data generator) is included in `artifacts/paper_1/`. Running `python generate_synthetic_finol.py` followed by `python run_baseline_benchmark.py` reproduces Tab 2, Tab 3, and the per-detector metrics within floating-point tolerance. Local Python output values are pre-computed and shipped in `artifacts/paper_1/results/` for reviewers without a Python environment.

Data availability

Synthetic data is regenerable from the seed 20260505. No real claims data is released. Carriers interested in promoting the artifact to Gold reproducibility tier are invited to contact the author.

Conflict of interest statement

The author declares no competing financial or non-financial interests in this work.

Author contributions

P.K.V. designed the feature schema, wrote the SQL view layer, implemented the Python benchmark harness, ran the experiments, analysed the results, and wrote the manuscript.

Funding

This work was conducted independently and received no external funding.

References

- [1] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. ICDM 2008*, 2008. DOI: 10.1109/ICDM.2008.17.
- [2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," in *Proc. ACM SIGMOD 2000*, 2000. DOI: 10.1145/342009.335388.
- [3] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001. DOI: 10.1162/089976601750264965.
- [4] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "COPOD: Copula-Based Outlier Detection," in *Proc. ICDM 2020*, 2020. DOI: 10.1109/ICDM50108.2020.00135.
- [5] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining* (pp. 413–422). IEEE. <https://doi.org/10.1109/ICDM.2008.17>
- [6] C. Zhou and R. C. Paffenroth, "Anomaly Detection with Robust Deep Autoencoders," in *Proc. ACM KDD 2017*, 2017. DOI: 10.1145/3097983.3098052.
- [7] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python Toolbox for Scalable Outlier Detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019. arXiv: 1901.01588.
- [8] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), Article 15, 1–58. <https://doi.org/10.1145/1541880.1541882>
- [9] L. Perini, V. Vercruyssen, and J. Davis, "Quantifying the Confidence of Anomaly Detectors in Their Example-Wise Predictions," in *Proc. ECML PKDD 2020*, LNAI 12459, pp. 227–243, 2020. DOI: 10.1007/978-3-030-67664-3_14.
- [10] J. M. Hellerstein et al., "The MADlib Analytics Library or MAD Skills, the SQL," *Proc. VLDB Endowment*, vol. 5, no. 12, 2012. DOI: 10.14778/2367502.2367510.
- [11] Ormenişan, A. A., Ismail, M., Hammar, K., Andersson, R., Gebremeskel, E., Kakantousis, T., Kouzoupis, A., Buso, F., Berthou, G., Dowling, J., & Haridi, S. (2019). Horizontally scalable ML pipelines with a feature store. In *Proceedings of the 2nd SysML Conference*
- [12] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," in *Proc. NeurIPS 2015*, 2015.
- [13] B. J. Heil et al., "Reproducibility standards for machine learning in the life sciences," *Nature Methods*, vol. 18, pp. 1132–1135, 2021. DOI: 10.1038/s41592-021-01256-7.
- [14] C. Phua, V. Lee, K. Smith, and R. Gayler, "A Comprehensive Survey of Data Mining-based Fraud Detection Research," arXiv: 1009.6119, 2010.
- [15] Matloob, I., Khan, S. A., & Rahman, H. U. (2020). Sequence mining and prediction-based healthcare fraud detection methodology. *IEEE Access*, 8, 143256–143273. <https://doi.org/10.1109/ACCESS.2020.3013962>
- [16] Subudhi, S., & Panigrahi, S. (2018). Detection of automobile insurance fraud using feature selection and data mining techniques. *International Journal of Rough Sets and Data Analysis*, 5(3), 1–20. <https://doi.org/10.4018/IJRSDA.2018070101>

- [17] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016. DOI: 10.1016/j.eswa.2015.12.030.
- [18] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, vol. 557, pp. 317–331, 2021. DOI: 10.1016/j.ins.2019.05.042.
- [19] A. E. Young, E. Besen, and J. Willetts, "The Relationship Between Work-Disability Duration and Claimant's Expected Time to Return to Work as Recorded by Workers' Compensation Claims Managers," *Journal of Occupational Rehabilitation*, vol. 27, no. 3, 2016. DOI: 10.1007/s10926-016-9656-z.
- [20] Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, 30. <https://proceedings.neurips.cc/paper/2017/hash/5d44ee6f2c3f71a6d8c7b6f6b3f8b5e5-Abstract.html>
- [21] Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2017). Data management challenges in production machine learning. *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17)*, 1723–1726. <https://doi.org/10.1145/3035918.3054782>
- [22] Breck, E., Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2019). Data validation for machine learning. *Proceedings of Machine Learning and Systems (SysML 2019)*.