

Label Provenance and Innocent Populations in Synthetic ERP Fraud Benchmarks

Labelled occupational fraud data is scarce, because identifying fraud inside real enterprise ledgers requires both known cases and expensive expert annotation.

Axolile Lungu

founder of [synthetic cfo](#)

9 September 2026 | 10 min read

Abstract

Labelled occupational fraud data is scarce, because identifying fraud inside real enterprise ledgers requires both known cases and expensive expert annotation. Synthetic generation is the common response. We identify a failure mode that makes many such benchmarks unable to measure the quantity they claim to measure. When ground truth is derived, that is, produced by querying the finished dataset for records satisfying a fraud condition, the label set becomes definitionally identical to the condition. No innocent instance of the condition can exist anywhere in the data, so a detector implementing that condition cannot produce a false positive, and precision is not measurable. We give three diagnostics that expose this property in any labelled dataset, report measurements from a production ERP generator before and after repair, and show that repairing it makes benchmark scores fall while ceiling recall holds or rises. We argue that this combination, lower precision with sustained ceiling recall, is the signature of a benchmark that has become measurable rather than one that has become noisy.

1. Introduction

Automated detection of occupational fraud in enterprise resource planning systems is held back by a data problem rather than a modelling problem. Real ledgers containing confirmed, adjudicated fraud are rare, commercially sensitive, and almost never shareable. The published literature repeatedly notes that ERP data is not publicly available for the development and comparison of detection methods, and that even unsupervised anomaly detection requires labelled data at evaluation time in order to select preprocessing, models and hyperparameters.

Synthetic generation is the natural answer, and several generators exist. The question this note addresses is not whether synthetic ERP data can be made realistic. It is whether a score measured on such data means anything, and specifically whether the way the answer

key is produced can silently guarantee a good result.

Our claim is that it frequently can, that the defect is mechanical rather than subtle, and that it is cheaply detectable with tests any consumer of a labelled dataset can run in a few minutes.

2. Recorded labels and derived labels

There are two ways to produce an answer key for a generated dataset.

Recorded. The generator plants a scheme and writes down, at that moment, exactly which records it touched. The label is a record of an act of construction.

Derived. A harvester queries the finished database afterwards and labels whatever the query returns. The label is a restatement of a predicate.

The distinction looks procedural and is not. Consider a phantom revenue scheme, in which invoices are raised to a customer who never pays. Derived, the label for that scheme might be written as the set of accounts receivable rows with no clearing date, which in SAP terms is approximately `WHERE AUGDT = ''`. Every open receivable in the dataset is now labelled fraud.

Three consequences follow immediately.

First, precision is not measurable for that condition. A detector that flags open receivables achieves perfect precision by construction, because the dataset contains no open receivable that is not labelled fraud. The detector cannot be wrong. A benchmark on which a detector cannot be wrong does not measure detection.

Second, the difficulty of the benchmark is misrepresented. In our own system before repair, a two line textbook rule reached grade C on the SAP world and grade A on the Oracle world with no forensic reasoning of any kind.

Third, and most seriously for downstream use, a model trained on the data learns a false rule. It learns that an open receivable indicates fraud. In every real ledger, most open receivables are simply not yet due.

3. Three diagnostics

The following tests require only the dataset and its answer key. They do not require access to the generator.

3.1 The condition satisfaction test

For each label family, write the plainest query that selects the records that family describes. Count the selected records that are **not** labelled fraud.

If the count is zero, the answer key is a restatement of the query, and any score on that family measures the generator rather than the detector.

Applied to our own system before repair, across eight SAP conditions covering 212 of 239 labels and four Oracle conditions covering 194 of 272 labels, zero rows satisfied a fraud condition without being labelled fraud.

3.2 Single field separability

For each fraud vector, test whether any single field separates the labelled population from the rest of the dataset. A vector separable by one field is learnable without any understanding of the underlying scheme, and a model that learns it has learned the generator's shortcut.

Before repair, 4 of 15 SAP vectors and 6 of 13 Oracle vectors were separable by a single field.

3.3 Marker column leakage

Generators commonly carry provenance columns that are not part of the target system's data dictionary, for example scenario tags, boolean scheme flags, or annotation text whose fraud variants begin with a distinctive token. If these columns ship inside the evidence a model is given, the answer key is present in the input.

We measured this directly. With marker columns left in place, a two line regular expression scored grade C on the SAP world, at 82 percent recall, and grade B on the Oracle world, at 90 percent recall with nine of thirteen vectors fully recovered, with no forensic work at all.

4. Repair

The repair has two halves, and neither works alone.

4.1 Record, do not derive

Every generator appends the labels for what it plants at the moment it plants it, in the answer key row shape, comprising module, control identifier, fraud vector, source table, record identifier and detail. The orchestrator folds these lists, sorts them for byte stability, and writes a single ground truth file. The file is always written, header only in a clean build, so that the package contents and its hash manifest always match the accompanying documentation.

Recording alone is insufficient. If the generated world still contains no innocent instances of a fraud condition, the recorded labels remain coextensive with the condition even though they were not derived from it.

4.2 Plant the innocent population

The world is framed as a fiscal year extract taken during the following close, with an extract cutoff after year end. Anything that would settle after the cutoff is simply open, exactly as December invoices on 45 or 60 day terms are open in any real year end extract.

Nine innocent populations were added, each neutralising the surface of one fraud condition. They comprise open receivables that are merely not yet due, cheques voided for a misprint and reissued, a non purchase order expense desk covering rent and utilities, backorders billed at received value, error keyed same reference reversals, year end audit adjustments posted past the 45 day mark, routine manual journals and reserve movement, weekend batch postings, and deposits in transit. Later engine versions added lifecycle and banking populations.

The design rule is that each innocent population must be a genuine feature of real ledgers, not a decoy inserted to defeat the diagnostic. If the population would not occur in a real company, planting it degrades realism in exchange for a better looking metric, which is the same error in the opposite direction.

4.3 Withhold the answer key from the evidence

Marker columns are dropped from every table in the database and from the source workbooks, and any view built on them is dropped with them. A verifier re-scans the shipped package and fails the job if a single marker survives.

5. Results

All figures are measured on the same generator. The before column is engine version

1.7.0. The after column is engine version 1.10.2, standing unchanged on 1.10.3.

Metric	Before	After
Innocent instances of fraud conditions	0 SAP, 0 Oracle	129 SAP, 106 Oracle
Vectors separable by a single field	4 of 15 SAP, 6 of 13 Oracle	0 SAP, 1 Oracle
Textbook rule precision	0.69 SAP, 0.98 Oracle	0.41 SAP, 0.61 Oracle
Textbook rule grade	C SAP, A Oracle	C SAP, C Oracle
Ceiling recall	0.92 SAP, 0.93 Oracle	0.995 SAP, 0.992 Oracle

Two observations matter more than the individual figures.

Precision fell. On the Oracle world a textbook rule set moved from 0.98 to 0.61, and its grade from A to C. Read naively this is a regression. It is the intended result. Precision could not previously be low, because there was nothing in the data for the rule to be wrong about.

Ceiling recall held or rose, from 0.92 to 0.995 on SAP and 0.93 to 0.992 on Oracle. The ceiling is a rule set written with full knowledge of which schemes were planted, and it represents the best achievable score rather than a realistic one. That this figure did not fall is the validity argument. Had the repair worked by making labels arbitrary or unsupported, the ceiling would have fallen with everything else. Instead, every labelled record remains recoverable from evidence present in the data, while the population of innocent look alikes now makes recovering them without false positives genuinely difficult.

We suggest that this pattern, falling precision with sustained ceiling recall, is the signature that distinguishes a benchmark repair from benchmark noise, and that authors reporting such repairs should be expected to show both figures.

5.1 Detector tiers after repair

For context, three detector tiers scored against the repaired SAP world, which carries 206 labelled records.

Detector	Precision	Recall	F1	Grade	Flags on a fraud free twin
Generic statistical anomaly detection	0.067	0.218	0.102	F	596
Textbook rule set	0.410	0.718	0.522	C	210

Detector	Precision	Recall	F1	Grade	Flags on a fraud free twin
Targeted forensic rules, ceiling	0.438	0.995	0.608	C	277

The fraud free twin is the same world regenerated from the same seed with fraud disabled and an empty answer key, so every flag raised against it is a false positive by construction. We note this as a second instrument that generated data makes available and real data does not, since no real organisation can warrant a ledger as containing no fraud.

The failing grade for generic statistical anomaly detection is worth stating plainly rather than burying. The planted schemes are behavioural rather than large in magnitude, so outlier tests, digit distribution tests and round number screens do not engage with them.

6. Conditions that remain closed, and why

Two conditions in our system remain closed sets, meaning every instance of the condition is labelled fraud. These are segregation of duties violations, where one user both raises a purchase order and posts the corresponding invoice, and ghost vendors, where a vendor bank account matches an employee bank account.

We report these rather than repair them, because in these two cases the join is the analytic. A user who performs both sides of that transaction is a genuine control exception every time, and a vendor account matching an employee account is a genuine exception every time. Planting innocent instances would not improve realism, it would introduce records that are wrong.

The general principle is that the condition satisfaction test is a diagnostic and not a target. Where a condition is genuinely closed in the real world, a closed label set is correct, and the appropriate response is disclosure rather than repair. We therefore report these as recorded closed with the reason attached, so that a consumer of the dataset encounters the fact rather than discovering it.

7. Limitations

Human resources labels in our system are still harvested by tag rather than recorded into the plant time list. We consider this non tautological, because the tag records the act of planting rather than restating a query, but it is a weaker guarantee than the recorded path and we state it as such.

One single field separable vector survives on the Oracle world, a missing trader scheme in which three invoices originate from one vendor and are therefore separable by that vendor identifier. This is inherent to a single perpetrator scheme rather than a leak, but we report it rather than excluding it from the count.

The measurements in this note come from one generator. The diagnostics in section 3 are general, but the effect sizes are not, and we would expect them to vary substantially with the scheme mix and the realism of the surrounding world.

Finally, a benchmark measures detection on generated data. It does not establish that a detector performing well here will perform well on any particular real ledger. What it establishes is a lower bound on rigour, in that a detector which cannot pass a benchmark where the answer is known should not be trusted where it is not.

8. Reproducibility

Every figure reported here regenerates from a printed seed. The generator is deterministic, with no language model involved in producing any financial figure, and each package ships a certificate listing the SHA-256 digest of every file it contains, scoped to engine version, seed and configuration. The same three inputs reproduce the package byte for byte.

The scoring harness, the condition satisfaction counts and the separability counts are re-runnable rather than reported only in summary.

9. Conclusion

The provenance of a label determines whether a score computed against it means anything. Derived ground truth in synthetic fraud data produces benchmarks on which detectors cannot fail and models learn false rules, and it does so invisibly, since the resulting datasets look rigorous and produce impressive numbers.

We recommend that publishers of labelled synthetic datasets report, as standard, the count of records satisfying each fraud condition without carrying its label, the number of vectors separable by a single field, and confirmation that provenance markers are absent from the evidence given to a model. Consumers of such datasets can compute the first two themselves in minutes, and we would encourage them to do so before trusting any published score.

Cite this

Axolile Lungu (2026). Label Provenance and Innocent Populations in Synthetic ERP Fraud Benchmarks. synthetic cfo.

<https://app.syntheticcfo.com/articles/label-provenance-and-innocent-populations-in-synthetic-erp-fraud-benchmarks>

```
@misc{lungu2026label,  
  author = {Axolile Lungu},  
  title = {{Label Provenance and Innocent Populations in Synthetic ERP Fraud Benchmarks}},  
  year = {2026},  
  month = sep,  
  publisher = {synthetic cfo},  
  url =  
{https://app.syntheticcfo.com/articles/label-provenance-and-innocent-populations-in-synthetic-erp-fraud-benchmarks},  
  howpublished =  
{url{https://app.syntheticcfo.com/articles/label-provenance-and-innocent-populations-in-synthetic-erp-fraud-benchmarks}}  
}
```

This copy was set from the page at

<https://app.syntheticcfo.com/articles/label-provenance-and-innocent-populations-in-synthetic-erp-fraud-benchmarks>. The page is the version of record.