findic

WHITE PAPER

Anomalies Without Guidance

Self-supervised anomaly detection in transaction and claims data

Contents

About this white paper

The detection of fraud and money laundering in transaction and claims data looks at first glance like a classical modelling problem: identify suspicious cases and assess risks. In practice the task is considerably harder. Confirmed fraud cases are rare, existing labels are often biased, rule sets age quickly, and many suspicious patterns arise only in the interplay of accounts, contracts, payment flows and temporal sequences. That is where simple thresholds and isolated single models reach their limits.

This white paper describes how our anomaly detection analyses such risks from several perspectives, namely through transaction-level features, graph structures, sequences and traceable rules. At its centre is not the promise of an automatic fraud decision but a robust prioritisation of cases for review. It aims to show which modelling decisions are necessary for that, why genuine scores matter more than seemingly precise probabilities, and where the limits of such systems lie in operation.

1 · The label problem nobody admits

Fraud detection is readily treated in the literature as a classification problem. Take labelled transactions, train a model, measure AUC. In practice the starting position almost always looks different. This is not a marginal phenomenon but the well-known fundamental problem of supervised methods:[1] reliable labels simply do not exist. What exists instead is outdated, biased, and shaped by a reporting system in which "fraud" means what a case handler suspected at a particular point in time. Confirmed cases are rare; one per cent is already a lot. You only find the patterns you searched for.

Projects sidestep this in different ways. Three of them are dead ends.

The rule path. Rule-based systems create an impression of traceability and auditability. Yet they are blind to everything that was not yet known when the rule was formulated. Rule sets age badly. They answer yesterday's questions with growing precision.

The pseudo-label path. Take an unsupervised model, declare the top five per cent of its scores to be "fraud", and train a second, supervised model on them. At first this reads like a pragmatic detour, but it creates a circular argument as soon as the pseudo-labels are treated as ground truth without verification. The second model learns to reproduce the opinion of the first. With human counter-checking, confidence filtering or iterative relabelling, this approach can be rescued. Without such corrections, every systematic blindness of the first model is not corrected but amplified. Afterwards the evaluation looks good, because model two "predicts" what model one already prescribed. We have found this design in two third-party codebases, including a calibration on the pseudo-labels that gave the scores the appearance of genuine probabilities. That is how a certainty can arise in practice that does not hold up professionally.

[1] Hilal, W., Gadsden, S. A. & Yawney, J. (2022), "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances", Expert Systems with Applications 193

The synthetic path. Instead of inventing labels, one invents the cases themselves. Interpolation methods such as SMOTE generate artificial minority cases along the connecting lines of known fraud cases,[2] generative models such as CTGAN learn the distribution and sample new transactions.[3] This addresses the rarity but not the bias: anyone who computes a thousand new cases from a hundred known ones multiplies the investigation history instead of expanding it. Interpolation moreover knows no domain logic. It mixes fraud types into cases that never existed, and the generated points demonstrably often fall into regions of the majority class to which they do not belong.[4] With generative models the picture worsens. The most recent benchmarks show that synthetic transaction data can look deceptively real statistically while destroying exactly the behavioural structures that fraud detection depends on.[5] A model that shines on such data has learned the assumptions of the generator, not the patterns of the perpetrators.

Our path is more uncomfortable, but worth it: self-supervised methods that learn from the structure of the data itself what is normal, and that deal with the fact that their outputs are ranks, not probabilities. This is where findic comes in. Anyone who has no labels must not feign probabilities, and anyone who has them must reserve them for calibration and evaluation rather than training.

First, though, the terminology, before matters turn technical. We call unsupervised every method that manages without an external target variable. We call self-supervised the subset of these that constructs its training signal from the data itself. The masked transformer and the contrastive encoder are self-supervised, the isolation forest is unsupervised, and the rule view is not a learning method at all. What is scored is always a unit: depending on the project, the transaction or the claim. Nodes in the graph are accounts or contracting parties; edges are payment flows or substantiated similarity relationships. When we say "case" in this paper, we mean the scored unit of the respective project.

2 · Methods we thought were enough

Before we come to the architecture, a few experiences from the engine room. They explain why the system looks the way it looks.

Isolation Forest: faithful, but deaf to context

The isolation forest[6] is the workhorse of every anomaly pipeline, and it remains in our ensemble. That is a deliberate choice, and for good reason. Its limit is conceptual. It sees every transaction in isolation. An amount of €9,800 is either suspicious to it or not. That the same customer already moved €9,700 and €9,850 twice in the previous week escapes it, because it lacks any form of memory. Structuring, the classic money laundering pattern below the reporting threshold, exploits precisely this blindness, or more precisely: the blindness of the feature space. Anyone who explicitly encodes velocity counters,

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002), "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research 16, pp. 321–357

[3] Xu, L., Skoularidou, M., Cuesta-Infante, A. & Veeramachaneni, K. (2019), "Modeling Tabular Data using Conditional GAN", NeurIPS 2019

[4] Tarawneh, A. S. et al. (2022), "Stop Oversampling for Class Imbalance Learning: A Critical Review", IEEE Access

[5] Sajja, B. (2026), "Synthetic Tabular Generators Fail to Preserve Behavioral Fraud Patterns: A Benchmark on Temporal, Velocity, and Multi-Account Signals", arXiv:2604.13125

[6] Liu, F. T., Ting, K. M. & Zhou, Z.-H. (2008), "Isolation Forest", Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), pp. 413–422

cumulated amounts and distances to thresholds as features makes structuring visible to the isolation forest as well. Only then, these features have likewise been formulated by a human.

Gradient boosting

It is by now well documented that tree-based methods beat deep learning architectures on typical tabular data, systematically, across dozens of datasets and generous hyperparameter budgets.[7] Our experience agrees, with one refinement. On transaction-level features (amounts, time intervals, counters, ratios), LightGBM is our strongest single model[8] as soon as any labels exist. For such data, a deep network as the main model is therefore hard to justify unless the additional context is explicitly used.

But boosting knows neither neighbourhood nor order. There is no tree structure that expresses "this beneficiary hangs on a ring of 14 nearly identical accounts" without someone having formulated that pattern as a feature beforehand. That is where graph and sequence models come in.

Graph autoencoder: the DOMINANT legacy

The idea of finding anomalies via the reconstruction error of a graph autoencoder has been established since DOMINANT (SDM 2019):[9] a graph convolution encoder embeds every node, two decoders reconstruct attributes and adjacency respectively. Where both go badly, that is where the anomaly lies. Variants such as AnomalyDAE decompose attribute and structure space even more cleanly.[10] We have adopted and extended this basic pattern: attention instead of convolution (graph attention networks allow the neighbourhood to be weighted by relevance)[11], several relations in parallel (feature similarity, community proximity, real payment flows) with learnable weighting, and, this is the part barely described in the literature, an error combination that normalises with median and MAD instead of mean and standard deviation. Reconstruction errors are typically heavy-tailed; robust scaling prevents a few extreme values from dominating the entire score scale.

What we learned in the process: the graph is itself a modelling assumption. In an early version, because of a sorting error that went undiscovered for months, our k-nearest-neighbour graph connected nodes to their most dissimilar instead of their most similar neighbours. The system nonetheless delivered plausible scores, because autoencoders train with astonishing fault tolerance. Only during an inspection pass through the edge list did we notice the error. Since then, the rule with us has been: every graph construction is code that is reviewed just like the model. And every relation we span must be substantively justified. Our "mule relation" connects accounts that resemble each other conspicuously on the highest-variance features, because that is the pattern with which money mule networks operate in money laundering.[12] Concretely: we determine the features with the highest variance in the respective portfolio, standardise them robustly, and connect accounts via a k-nearest-neighbour graph on these features. Feature count and neighbourhood size are documented project parameters. We do not publish

---

[7] Grinsztajn, L., Oyallon, E. & Varoquaux, G. (2022), "Why do tree-based models still outperform deep learning on typical tabular data?", NeurIPS 2022

[8] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T.-Y. (2017), "LightGBM: A Highly Efficient Gradient Boosting Decision Tree", Advances in Neural Information Processing Systems 30

[9] Ding, K., Li, J., Bhanushali, R. & Liu, H. (2019), "Deep Anomaly Detection on Attributed Networks", Proceedings of the 2019 SIAM International Conference on Data Mining (SDM 2019)

[10] Fan, H., Zhang, F. & Li, Z. (2020), "AnomalyDAE: Dual Autoencoder for Anomaly Detection on Attributed Networks", Proceedings of ICASSP 2020

[11] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. & Bengio, Y. (2018), "Graph Attention Networks", ICLR 2018

[12] Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H. & Yu, P. S. (2020), "Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters", Proceedings of CIKM 2020

the full specification of the edge weights. What matters to us here is the principle. Every relation is code with a substantive justification, not a configuration that "somehow seemed sensible".

A second detail from the engine room belongs here deliberately, and not with the isolation forest: scaling. We used robust scalers (median, IQR) until a dataset with amounts spanning a factor of 100 showed that "robust" does not mean "bounded". After the transformation, the extreme values continued to dominate every distance and thus every k-nearest-neighbour graph. In our dataset, switching to a quantile transformation to a normal distribution proved more favourable, because it bound all features into comparable ranges. The transformation compresses precisely the tails of the distribution, and in fraud data the signal often sits in those tails. Grinsztajn and colleagues use quantile transformation to a normal distribution as preprocessing in their large tabular benchmark.[13] Our observation points in the same direction, and our experience shows that it also matters for distance-based graph construction. For the isolation forest, by contrast, the question is largely irrelevant. Its splits rest on orderings, and monotone transformations change nothing about orderings.

Sequence models: the transformer that learned bookkeeping

The most recent extension is a transformer over the transaction history of each account. Training follows the masking principle that BERT established for language.[14] A random part of the features of a random subset of transactions is hidden; the model reconstructs it from the context before and after. A model that has internalised the pattern of an account can reconstruct well. An account that suddenly departs from its behavioural pattern produces high errors. With that, the sequence view supplies the signal the isolation forest lacks: a memory. One refinement that matters in operation. Bidirectional reconstruction uses context before and after. That is permissible if reviews are retrospective, that is, all transactions of a completed period are scored at once. For real-time scoring the model must not see the future; in this operating mode we mask causally and use only the preceding context. Both modes are implemented, and we declare which one is running according to the deployment scenario.

From operation came two important observations. First: on unscaled raw values, the reconstruction loss on the first run rose to values beyond 10^{19}, because the squared error on euro amounts mocks every normalisation. Second: we cover the last 32 events per account. That represents a compromise between memory depth and padding share for accounts with little activity, not the result of a systematic hyperparameter search. Fast fraud stretching over weeks escapes this window logic and remains the graph's business.

Contrastive pre-training: the way out of the circular argument

Recall the pseudo-label trap from Section 1. The modern answer to it is contrastive learning: instead of inventing labels, the encoder learns that two slightly perturbed variants of the same transaction belong together (feature dropout here, Gaussian noise there) and differ from all other transactions. On graphs, this family has shown with methods such as SL-GAD that it makes anomalies visible without a single label by combining generative and contrastive signals.[15] We pre-train the GAT encoder with an NT-Xent

[13] Grinsztajn, L., Oyallon, E. & Varoquaux, G. (2022), ibid.

[14] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019), "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Proceedings of NAACL-HLT 2019

[15] Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K. T. & Chen, Y.-P. P. (2021), "Generative and Contrastive Self-Supervised Learning for Graph Anomaly Detection", IEEE Transactions on Knowledge and Data Engineering

loss function[16] before the actual autoencoder is trained. The effect compared with the old pseudo-label pipeline is qualitative. The representations are smoother, the scores more stable against the choice of graph, and nobody had to invent cases.

3 · The architecture: four views, one decision

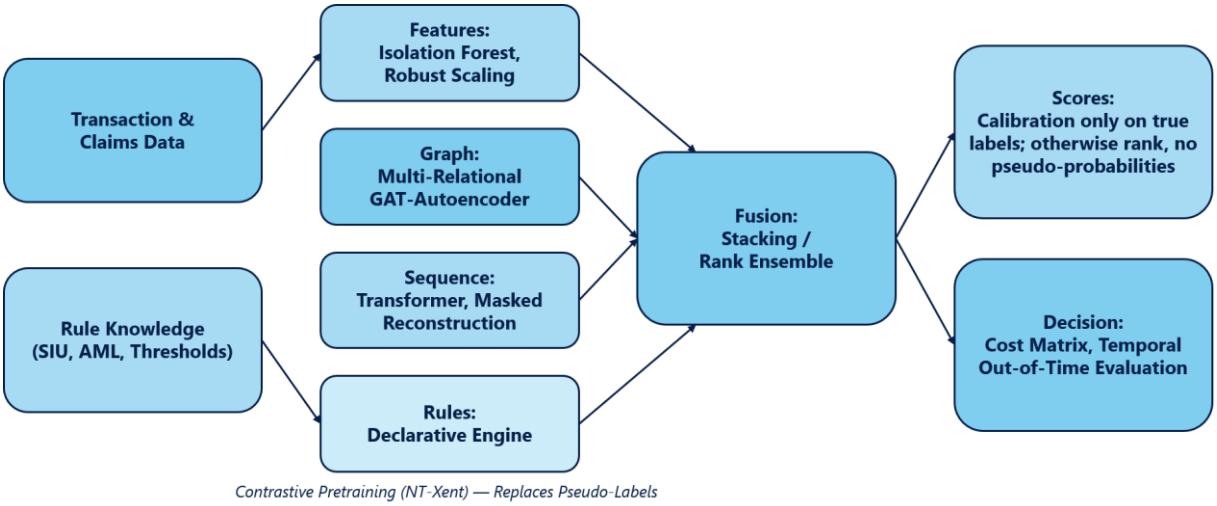

Figure 1: The four views of findic anomaly detection on a transaction. No single method is responsible for the overall statement; the fusion decides.

Four views meet, and none may decide alone: the attribute view (isolation forest on robustly scaled features), the graph view (the multi-relational GAT autoencoder), the sequence view (the masked transformer), and the rule view, deliberately not abolished, because rules are the only module that takes in domain prior knowledge directly and auditably. Fusion happens as stacking. If genuine labels exist, a LightGBM learns the combination. If none exist, ranks are averaged deterministically: seven rankings, none of them random, and no meta-learner deceiving itself.

Two decisions in this picture deserve a second look, because they go against convention.

We normalise via ranks, not via per-batch z-scores. The reason lies in the fusion: ranks make the differently scaled scores of the views combinable at all, and they are robust against monotone distortion of individual score scales. One restriction we state openly. Ranks are relative to the respective population. If the data stock changes, a customer's rank can move without his behaviour having changed. That is the same criticism that rightly applies to z-scores re-estimated per batch. Anyone who needs temporally stable thresholds cannot avoid a fixed reference distribution and calibration on confirmed feedback. That is where our ramp ends.

We dispense with interspersed random noise. In the codebase from which findic's anomaly detection emerged, there was a fallback that deterministically replaced degenerate score distributions with random numbers. That is practical in that the pipeline never crashes. It is at the same time problematic, because scores were no longer reproducible. Reproducibility given the same input is a central

[16]Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. (2020), "A Simple Framework for Contrastive Learning of Visual Representations", Proceedings of ICML 2020

precondition for an auditable pipeline, alongside the versioning of model, data, features, graph, parameters and thresholds. In some places this means letting an error become visible instead of concealing it behind a seemingly valid score distribution.

4 · With and without answers: the two operating modes

The first question in every initial meeting is: "Do you have labels?" The better question would be: "What would you do with them?" Because supervised and unsupervised are not camps between which one chooses in anomaly detection, but two operating modes of one and the same pipeline, with different strengths, different failure patterns, and a bridge between them. This chapter sets out what really changes when the answers meet the data.

What supervision brings and what it costs

Supervised learning brings three advantages that unsupervised learning cannot deliver. First, **probabilities**: only those who know what actually was fraud can measure scores against observed frequencies and calibrate them. Second, **measurability**: PR-AUC, precision@k, cost-optimised thresholds. All the metrics in the next chapter presuppose that there is a truth to compute against. Third, **explainability**: with labels, our stacking LightGBM trains on the seven base scores, and SHAP[17] tells us which of these input scores carried the meta-model's decision. That is an explanation of the model decision, not a causal explanation of the case. It is the difference between "the model says" and "here is why".

The price is rarely discussed. Labels in fraud data are not ground truth; they are a **case history**. They document what investigators searched for and what they found, not everything that existed. An SIU label stock is the map of the previous investigation strategy. A model trained on it learns the past of the investigations, not the present. On top of that comes ageing. What was confirmed in 2023 describes patterns of 2022. On such data, supervised learning shines at recognising known patterns again. It is at a disadvantage where perpetrators have changed their behaviour.

What self-supervision sees and what it does not

The unsupervised mode reverses the balance. Reconstruction errors, isolation scores and contrastive representations know no investigation history and can therefore point to patterns nobody ever searched for. The mule ring from Section 2 was such a case. The blindness lies elsewhere: without labels there is no probability, no recall figure and above all **no quick answer to the question "does it work?"**. The answer arises only in operation, case by case, through the review of the top ranks.

And the unsupervised mode has its own well-known failure pattern: the flood of legitimate unusualness. Large policyholders with thirty contracts, seasonal peaks in annual business, reconstruction cases after storms. In previous deployments, the upper risk list in the first month consisted of roughly half such

[17] Lundberg, S. M. & Lee, S.-I. (2017), "A Unified Approach to Interpreting Model Predictions", Advances in Neural Information Processing Systems 30 (NeurIPS 2017)

anomalies. Dealing with them is not a modelling question but a communication and feedback question. Every false alarm argued away must flow back into the system, or it will be reported again next week.

The graph as referee between the two worlds

The graph view of all things shows the difference between the operating modes most sharply. In unsupervised mode the GAT autoencoder is **judge and witness at once**. It defines via reconstruction what is normal, and it convicts deviations. That is powerful and risky, because whoever builds the graph wrongly builds normality wrongly, and nobody notices, because there is no label to object. In supervised mode the same graph encoder is demoted to **witness**. Its scores become one of seven features in the stacking, and the meta-model decides on the basis of genuine cases how much its testimony is worth. Observation from practice: the graph score is almost always among the two most heavily weighted SHAP drivers.

We deliberately do **not** train the graph itself in a supervised fashion, although that would be technically possible. For direct classification on graphs, confirmed case numbers, rarely more than a few hundred, are too little substance for a network with thousands of parameters. The result would be investigation history learned by rote. The division of labour has proved itself. The graph learns the geography of normality self-supervised; the small, fast boosting model on top learns the decision supervised. Anyone who wants to extract more from the few positives in this field methodologically ends up, incidentally, at PU learning, learning from positive and unlabelled data, a literature we follow attentively but do not yet consider mature for production use.[18]

The bridge: feedback becomes supervision

Between the two operating modes there is no jump but a ramp. It is part of the product. Investigators mark confirmed hits and false alarms in daily work. The feedback is stored via stable transaction IDs, not via row positions. We learned that after a re-import once shifted all markings onto the wrong cases. Calibration switches from rank to probability output as soon as enough confirmed cases exist to estimate a calibration curve with defensible uncertainty. We set the minimum quantity on a project basis, by case count, base rate and required precision, not by a blanket rule. Supervised stacking follows as soon as out-of-time validation shows that the meta-model actually beats deterministic rank averaging. The only thing that changes is the decision layer. For customers this means: starting without labels is not a dead end with an expiry date but the first floor of the same building. We distinguish four phases of one and the same pipeline: *label-free discovery* (no feedback, pure ranking), *feedback-driven ranking* (investigator markings flow back), *calibrated ranking* (probabilities on confirmed feedback), and *supervised fusion* (meta-model on genuine labels). The architecture remains identical across all phases; what matures is solely the decision layer.

[18]Elkan, C. & Noto, K. (2008), "Learning Classifiers from Only Positive and Unlabeled Data", Proceedings of KDD 2008

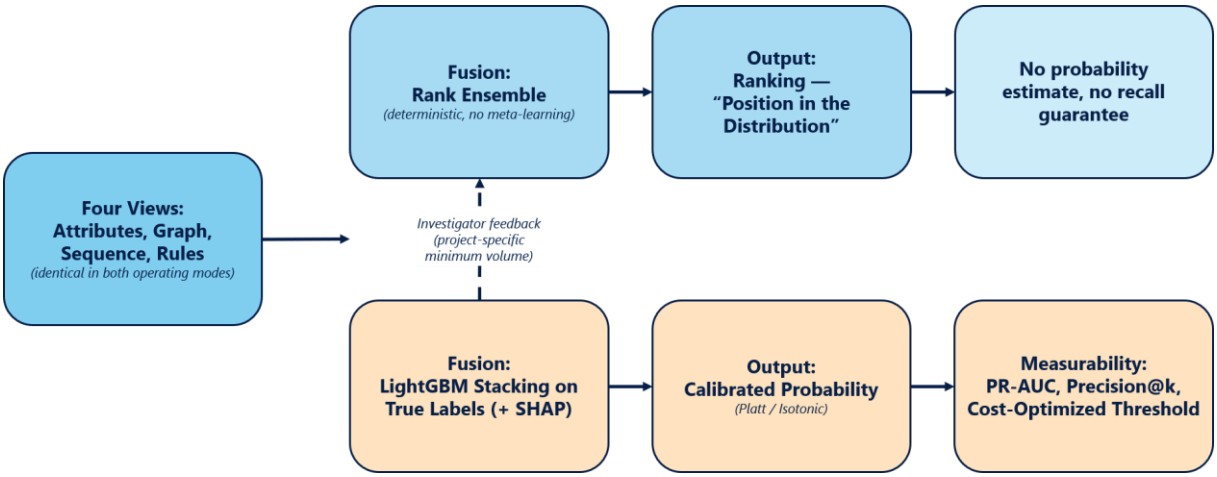

Figure 2: Two operating modes, one pipeline. The branching lies exclusively in the fusion layer; investigator feedback forms the ramp between them.

Dimension	Without labels (unsupervised)	With labels (supervised)
Output	Ranking, declared as ranks	Calibrated probability
Measurability	Only in operation, via review results	Immediately: PR-AUC, precision@k, ECE
Strength	Finds patterns never searched for	Reliably recognises known patterns again
Typical error	Flood of legitimate unusualness	Repeats investigation history
Vulnerability	Graph construction, seasonality	Label bias, pattern ageing
Communication	"Review the top 50, with us"	"The top 50 contain 40 fraudsters"
SHAP / explainability	Feature contributions to the reconstruction error	Full contribution analysis

Table 1: Supervision and self-supervision in direct comparison: strengths, limits and practical conditions of use.

The last row of this table deserves emphasis, because it is done wrongly again and again. An unsupervised model can very well explain which features were involved in the reconstruction error of a case. What it cannot do is say how likely fraud is. Anyone who throws both into one pot in a presentation is showing calibration without calibration data.

Our working principle is therefore simple. We do not ask "supervised or unsupervised?", but "where on the ramp do you stand today, and how fast can you move on?" Both with the same data and the same codebase.

5 · Measuring without deceiving yourself

The part where anomaly projects really fail is rarely the model; it is usually the evaluation.

Time beats randomness. The usual random split, 80 per cent training, 20 per cent test, does not measure generalisation on transaction data but memory. Fraud comes in waves, campaigns and ring closures. If the test period contains the same ring as the training, the model has already seen the answer. findic anomaly detection therefore splits exclusively along the time axis: training on the past, an embargo gap, testing on the future. The metrics that come out are lower than those from random splits.

PR first, ROC alongside. On data with one per cent positives, the ROC curve quickly looks better than operations are, because the large mass of correctly classified normal cases suppresses the false positive rate. The precision-recall curve is therefore the more informative instrument on imbalanced data.[19] We report PR-AUC as the primary metric and ROC-AUC as a supplementary one. On top comes precision@k for the list lengths an investigation team can realistically work through: 50, 100, 500 cases. For communication with business functions we add lift@k: the hit density of the top k relative to the base rate. "The top 1 per cent list contains eight times as many confirmed cases as a random sample" is the sentence that lands in a review committee, not the third decimal of an AUC. A ROC-AUC of 0.95 is useful for operations only if a workable list also emerges from it.

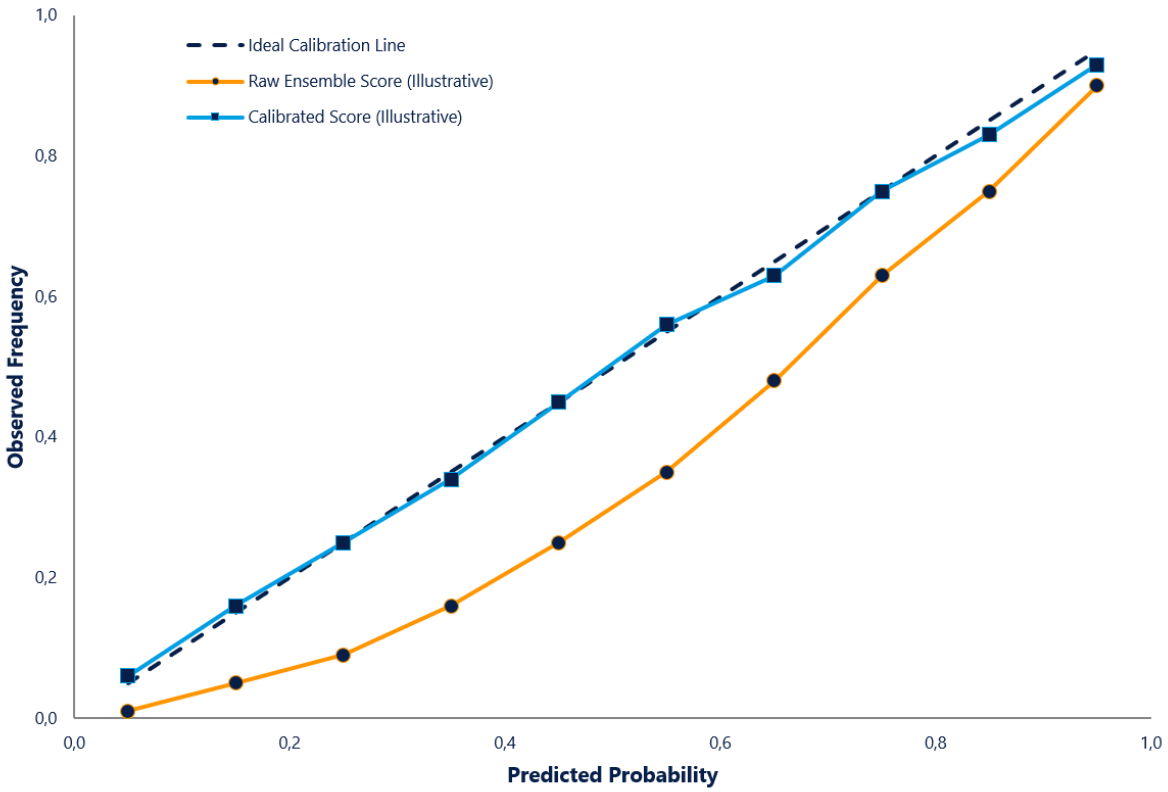

Figure 3: Illustrative reliability diagram. Calibration is not a cosmetic step but the precondition for thresholds and budgets being allowed to hang on scores. (The display illustrates the principle; no customer data.)

[19] Saito, T. & Rehmsmeier, M. (2015), "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets", PLOS ONE 10(3): e0118432

Calibrated. Probabilities have value only if they measurably keep what they promise. This is measured with the expected calibration error (ECE). Cases are sorted by their score and divided into equally sized groups (bins). In each group, the mean predicted probability is compared with the actual share of confirmed cases. The ECE is the mean deviation weighted across the groups. Zero would be perfect calibration. Modern networks are known to be systematically overconfident, and a one-parameter post-hoc adjustment is often enough to bring them back into line.[20] Our rule is harder than the state of the art. Calibration happens only on genuine labels or on confirmed investigator feedback. Where neither is available, findic anomaly detection outputs ranks and says that they are ranks. We have resisted the temptation to cast rankings into sham probabilities between 0.01 and 0.99 with a min-max stretch since the day such a stretch appeared in a management presentation in an earlier project as a "fraud probability".

The threshold is a controlling question. If a model flags 200 transactions but the team can review 40, the discussion about the third decimal of the AUC is over. We derive thresholds from cost matrices: what does an unnecessary review cost, what an overlooked loss? We derive the choice of threshold from the error costs and the expected base rate, not from a blanket default of 0.5. This goes back to Elkan[21] and in practice is still rarely implemented consistently.

6 · Putting the system to the test

A synthetic stress test, 600 generated transactions, 80 accounts, 40 injected cases with realistic patterns from both offence fields (high amounts on young policies, velocity bursts, a mule ring in a money laundering pattern), serves us as a regression test for every code change. There the pipeline separates the injected cases with an out-of-time ROC-AUC of 0.99 and a PR-AUC of 0.96. All 20 fraud cases of the test period stand within the top 50 of the risk list. This shows that the code works as intended across training, saving, loading and rescoring. It shows nothing about real data, on which patterns overlap, fields are missing and fraudsters adapt.

In one deployment, around 90,000 claims, three years of history, without confirmed labels, the expected and nonetheless gratifying picture emerged: the top 1 per cent of the risk list contained a visible concentration of cases the existing rule set knew as "borderline, not escalated", plus a small group of cases with community structures that could subsequently be assigned to an agency in an internal review, carried out by reviewers who did not know the model scores. Two of these cases had until then gone unnoticed in any review procedure. We deliberately name no percentages. Without confirmed labels and with a non-randomised selection of reviewed cases, every hit rate would be a sham precision. What this observation is worth is decided in accompanied operation with a feedback loop. The false positives in the same regions of the list were predominantly genuine but unusual events, such as large policyholders with many contracts, in other words the error type we expect and that can be argued away professionally.

[20] Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. (2017), "On Calibration of Modern Neural Networks", Proceedings of ICML 2017

[21] Elkan, C. (2001), "The Foundations of Cost-Sensitive Learning", Proceedings of IJCAI 2001, pp. 973–978

The next stage of development is half a year of accompanied operation with a feedback loop: investigators mark confirmed hits and false alarms, and only once enough feedback exists does the calibration switch from rank to probability output.

7 · Where are the limits?

Much of it hangs on the graph. The edges arise from keys, IDs and addresses, and if those are already inaccurate in the raw material (duplicates, addresses instead of persons, accounts that happen to be named the same), the system clusters constructs that do not exist at all. No model quality can correct a wrongly knotted world. That is why in every project more time flows into graph construction than into any model choice.

A further factor is time. Fraud patterns are under adaptive pressure: what stood out as structuring in 2024 is long since a different figure in 2026, and a model trained yesterday does not recognise tomorrow's patterns by itself. Anomaly detection without planned retraining and drift monitoring degrades.[22] We therefore document every model version with split parameters, thresholds and metrics in machine-readable form.

The next limit is conceptual. The system is predictive, not causal: a high score says “looks like the cases that ended badly” and not “is fraud”. Anyone who derives automated blocks from it instead of review orders risks decisions that are hard to defend professionally as well as regulatorily. Our system delivers prioritised review lists, and the decision remains with the human. For a long time we phrased this as a restriction; by now we see it as the only operating mode that withstands an audit.

And finally, fourth: substance beats architecture. Below a few thousand transactions or without temporal depth in the history, the graph and sequence views measure mainly noise. In that situation the honest recommendation is not the more complex model but classical methods, cleanly implemented. Plus the suggestion to build the data basis first and come back later.

8 · What a pilot with us means

Concretely: an export of the transaction or claims data, a few weeks of runtime, a professional contact person who knows the rule knowledge and review capacity. At the end stand a prioritised risk list with reasons per case and a written report on the chosen graph relations and threshold logics.

The question that is worth asking, in our experience, is not only “Do you have AI?”. More important is: “How do you make sure your evaluation is robust?”

[22] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. & Bouchachia, A. (2014), “A Survey on Concept Drift Adaptation”, ACM Computing Surveys 46(4)

9 · Sources and further reading

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16, 321–357. The reference method of synthetic oversampling; interpolates minority cases between their nearest neighbours.

Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. Proceedings of ICML 2020 (PMLR 119:1597–1607). SimCLR; origin of the NT-Xent loss function in our contrastive pre-training.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019. The masking principle behind our sequence view.

Ding, K., Li, J., Bhanushali, R. & Liu, H. (2019). Deep Anomaly Detection on Attributed Networks. Proceedings of the 2019 SIAM International Conference on Data Mining (SDM). DOMINANT, the blueprint of the graph autoencoder approach.

Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H. & Yu, P. S. (2020). Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters. Proceedings of CIKM 2020. CARE-GNN; fraud detection on graphs against perpetrators who deliberately camouflage themselves.

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. Proceedings of IJCAI 2001, 973–978. Why the default threshold of 0.5 is almost always the wrong business decision.

Elkan, C. & Noto, K. (2008). Learning Classifiers from Only Positive and Unlabeled Data. Proceedings of KDD 2008. PU learning: the methodological candidate for handling confirmed cases without reliable negatives.

Fan, H., Zhang, F. & Li, Z. (2020). AnomalyDAE: Dual Autoencoder for Anomaly Detection on Attributed Networks. Proceedings of ICASSP 2020. The cleaner decomposition of attribute and structure space towards which our graph view is oriented.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. & Bouchachia, A. (2014). A Survey on Concept Drift Adaptation. ACM Computing Surveys 46(4). The literature basis behind our drift monitoring.

Grinsztajn, L., Oyallon, E. & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? NeurIPS 2022, Datasets and Benchmarks Track. The evidence that boosting beats deep learning on tabular data.

Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. Proceedings of ICML 2017, 1321–1330. Overconfident networks and the surprising effectiveness of simple post-hoc calibration.

Hilal, W., Gadsden, S. A. & Yawney, J. (2022). Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. Expert Systems with Applications 193. Overview of anomaly detection specifically in financial fraud.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30.

Liu, F. T., Ting, K. M. & Zhou, Z.-H. (2008). Isolation Forest. Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), 413–422. The method behind our attribute view.

Lundberg, S. M. & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems 30. SHAP, the basis of our contribution analysis in supervised mode.

Saito, T. & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLOS ONE 10(3): e0118432. The standard argument against ROC-AUC with rare positive cases.

Sajja, B. (2026). Synthetic Tabular Generators Fail to Preserve Behavioral Fraud Patterns: A Benchmark on Temporal, Velocity, and Multi-Account Signals. arXiv:2604.13125. Current benchmark on the behavioural fidelity of synthetic data: CTGAN, TVAE, GaussianCopula and TabularARGN achieve good statistical fidelity and usable AUROC values, but destroy the behavioural structures fraud detection relies on: burst patterns, velocity rules, rings via shared devices (degradation by a factor of 17 to 100 against the real noise level); it also documents the minority class collapse of TVAE, in which the generated fraud rate drops from 3.5% to 0.03%.

Tarawneh, A. S. et al. (2022). Stop Oversampling for Class Imbalance Learning: A Review. IEEE Access. Systematic criticism: synthetic points demonstrably often fall into majority regions; oversampling as a "deceptive" approach.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. & Bengio, Y. (2018). Graph Attention Networks. ICLR 2018. The attention mechanics in our graph encoder.

Xu, L., Skoularidou, M., Cuesta-Infante, A. & Veeramachaneni, K. (2019). Modeling Tabular Data using Conditional GAN. NeurIPS 32. CTGAN: the reference method of generative table synthesis.

Zadrozny, B. & Elkan, C. (2002). Transforming Classifier Scores into Accurate Multiclass Probability Estimates. Proceedings of KDD 2002. Platt scaling and isotonic regression in comparison.

Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K. T. & Chen, Y.-P. P. (2021). Generative and Contrastive Self-Supervised Learning for Graph Anomaly Detection. IEEE Transactions on Knowledge and Data Engineering. SL-GAD, the template for combining reconstruction and contrastive signal.

10 · Contact

Udo Jacobasch
Partner
Udo.Jacobasch@findic.de
Office Berlin

Dr. Josef Dirnberger
Manager
Josef.Dirnberger@findic.de
Office München

Hamburg

Kurze Mühren 20,

D-20095 Hamburg

Phone +49 40/303740-0

E-Mail info@findic.de

Zurich

Gutenbergstrasse 1,

CH-8002 Zurich

Phone +41 44/56098-00

E-Mail info@findic.ch