FRAUD DETECTION: FINDING THE TRANSACTIONS A MODEL CANNOT JUDGE

Download

An illustration of how Squint Vision Studio maps a fraud model's decision space, separating the transactions it can assess reliably from the ones where its confidence is unearned.

The Benchmark | Why accuracy fails | The narrow band | Mapping the space | Graduated response | What changes

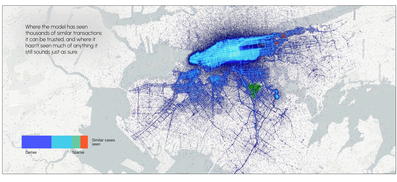
The most widely used public benchmark for card fraud came out of a research collaboration between the Machine Learning Group at Université Libre de Bruxelles and the payment processor Worldline. It contains 284,807 real card transactions made by European cardholders over two days, and 492 of them are fraudulent. That works out to 0.17 percent fraud, which means 99.83 percent of the data is legitimate. The features are anonymised through PCA into 28 numerical components, with transaction time and amount left in their original form.

Those proportions are what make this benchmark so useful because they expose a trap that catches teams in production. A model trained on this data can report accuracy near 99.8 percent while contributing almost nothing to fraud detection.

WHY THE ACCURACY NUMBER IS MEANINGLESS HERE

Consider a model that labels every transaction legitimate and never flags anything at all. On this dataset it would be correct 99.83 percent of the time because that is simply the share of legitimate transactions. The accuracy score rewards it handsomely for ignoring the rare class entirely, which is the one behaviour a fraud system can never afford.

Real models are more capable than that, but the same arithmetic hides their weaknesses. Their accuracy is dominated by the huge, easy legitimate majority, and it reveals nothing about the 492 cases that actually matter. This is why fraud research reports precision–recall metrics and recall on the fraud class rather than accuracy. The field has known for years that accuracy is the wrong instrument for this problem, yet accuracy remains the number that gets quoted when a model is signed off.



THE COSTLY CASES FALL IN A NARROW BAND

Fraud detection is unusual in that both kinds of error are expensive, and they pull against each other. A missed fraud is a direct financial loss. A false alarm blocks a paying customer, generates a support ticket, and chips away at trust, and false alarms tend to outnumber genuine fraud by a wide margin. Tuning the model to catch more fraud drives false alarms up, and tuning it to protect customers lets more fraud through.

The transactions that decide this trade-off are the ones where fraudulent and legitimate behaviour look nearly identical. In the ULB data these are the low-value, ordinary-looking transactions that cluster close to normal spending patterns, since fraudulent transactions in this dataset skew toward low values, with a median well below that of legitimate transactions. Nothing about them stands out from ordinary activity.

Fraud also has something most prediction problems lack, which is an opponent. Fraudsters study the defences and change their methods when a scheme is detected, so the patterns a model learned during training are gradually replaced by patterns designed to evade it. That new fraud arrives looking as ordinary as possible, which places it directly in the band where the model is already least reliable. The model gives no warning when this happens. A confident "legitimate" on an honest purchase and a confident "legitimate" on a novel fraud are indistinguishable at the output.

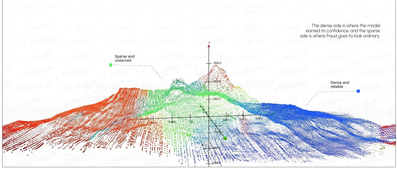
WHAT SQUINT VISION STUDIO ADDS TO THE MODEL

Vision Studio does not replace the fraud model, and it does not try to make it more accurate. It works alongside the model to establish where that model's judgment can be trusted.

The setup begins inside the model itself. Vision Studio can modify a model's computational graph to output the embeddings of any layer, which are the internal representations the model builds as it processes an input. Those embeddings are then reduced and visualised, so the model's decision-making space becomes something a developer can inspect directly rather than infer from scores.

What appears in that space is a structure the accuracy figure never showed. Ordinary legitimate transactions occupy dense, well-populated regions where the model has seen thousands of similar cases and its judgment is dependable. Vision Studio labels these the trusted regions. Elsewhere, fraudulent and legitimate transactions overlap, the data is sparse, and the model's separation between the two classes breaks down. These are the ambiguous regions, and in fraud they matter more than anywhere else because that is precisely where evasive fraud is designed to hide.

Slice-level analysis makes the same point in terms a fraud team can act on, by identifying which features and which regions of the data the model handles well and which it struggles with. Before deployment a team can see that the model is dependable on high-value anomalous transactions and unreliable on the small, ordinary-looking ones. That is the inverse of what the loss numbers would suggest.



A GRADUATED RESPONSE FOR REAL-TIME DECISIONS

Fraud decisions are made in milliseconds, at volume, so the response to an untrustworthy call has to be practical. Sending every doubtful transaction to a human analyst would be impossible at scale, and blocking them all would turn the model's uncertainty into blocked customers. The watchdog that Vision Studio generates and exports handles this by matching the response to the region the transaction falls in.

Transactions in the trusted regions are decided automatically at full speed. This is the great majority of traffic, and the throughput that makes automated fraud detection worthwhile is preserved.

Transactions in the ambiguous regions are treated differently because the model's confidence there has not been earned. Rather than approving them on a guess or blocking them on suspicion, the system routes them to step-up verification. A second factor, an identity challenge, or an additional check gathered in real time, lets the decision rest on evidence the model did not have. An honest customer in an unusual situation is asked to confirm rather than turned away, and a fraud built to look ordinary is stopped at the point where looking ordinary is no longer enough.

Transactions in the ambiguous regions are treated differently because the model's confidence there has not been earned. Rather than approving them on a guess or blocking them on suspicion, the system routes them to step-up verification. A second factor, an identity challenge, or an additional check gathered in real time, lets the decision rest on evidence the model did not have. An honest customer in an unusual situation is asked to confirm rather than turned away, and a fraud built to look ordinary is stopped at the point where looking ordinary is no longer enough.

The small number of high-value cases that verification cannot settle go to a human analyst, whose time is now spent on cases that genuinely need judgment.

The boundary earns its keep a second way once the system is running. Because fraudsters move their activity toward the model's weak spots, a rising share of transactions landing in the ambiguous regions is an early indication that the attack pattern is shifting. That shift appears while the change is still small, well before it grows large enough to register as falling accuracy, which gives the team room to respond before losses accumulate.

WHAT THIS CHANGES

On a dataset where 99.83 percent of transactions are legitimate, an accuracy score measures the part of the work that was never difficult. It says nothing about the 492 transactions the system exists to catch, and it stays reassuring right up until the losses appear.

Mapping the model's decision space replaces that single number with something operational. The model continues to decide the routine majority at full speed, the transactions it cannot genuinely assess are met with evidence rather than confidence, and the shape of the boundary itself warns the team when the adversary moves. For a system whose entire purpose is catching the rare and deliberately hidden case, the useful question becomes which of the model's own decisions deserve to be trusted. Overall correctness answers something else entirely.

This scenario is representative. It draws on the public European cardholder transaction dataset assembled through the ULB Machine Learning Group's fraud detection research with Worldline, and on the group's published work on unbalanced classification and concept drift in fraud detection, including Dal Pozzolo and colleagues' papers on undersampling and calibration and their practitioner-perspective study in Expert Systems with Applications. Squint Vision Studio's capabilities, its embedding extraction, decision-space mapping, slice-level insights, and exportable watchdog are described as designed. The routing behaviour described is what the approach does by design and is not a measured result from a specific deployment.