A study in two modules ,

Credit Ratings, *and the* fraud beneath *them.*

A machine-learning pipeline for predicting US public-firm creditworthiness and detecting financial misreporting , built on public XBRL data, 2009–2016.

AUTHORS

Avneet Kaur · Hunnar Khurana · Renaya Gupta

UNIVERSE

444 firms · 2,344 obs

SEC EDGAR XBRL · AAER · FRED

THE PROBLEM

Two questions, *one pipeline.*

Rating agencies sell creditworthiness as a single grade between AAA and D. Module 1 asks whether a model fit only on a firm's financials and macro environment can recover that grade. We treat it as ordinal regression on a 1 to 22 scale.

Module 2 inherits Module 1's residuals. When the predicted rating disagrees with the rating an agency actually assigned, that gap is an anomaly worth investigating. Combined with Beneish red flags and EDGAR event signals, it becomes a fraud classifier.

Eight scripts, forty minutes, one seed.

Q1

Can financial statements alone predict *creditworthiness*?

→ *Module 1, ordinal regression*

§

Q2

Can the *residuals* from that model flag financial fraud?

→ *Module 2, binary classification*

The bridge, Module 1's *(predicted minus actual)* rating gap becomes a Module 2 feature. One pipeline, two heads, sharing the residual.

WHERE THE FIELD STANDS

Six papers, *one gap.*

AUC consensus
0.65 – 0.725

PAPER	SAMPLE & SOURCE	METHOD	FINDINGS	LIMITATIONS
Beneish , M-Score 1999 · J. Fin. Econ.	74 manipulators + 2,332 controls Compustat, 1982–92	Probit on 8 accounting ratios , flag earnings manipulation	M-score > −1.78 = manipulator AUC ≈ **0.65 – 0.70**	Pre-SOX · single-period · high false positives · no restatement signal
Cecchini et al. 2010 · Management Science	132 fraud + matched controls AAER, 1991–2003	SVM with custom financial-ratio kernel	Beats standard SVM AUC ≈ **0.70 – 0.73**	Small fraud sample (N=132) · kernel hard to interpret · no real-time signals
Dechow et al. , F-Score 2011 · Contemp. Acc. Res.	2,190 AAER firm-years Compustat, 1982–2005	Logistic regression , accruals, performance, market, off-B/S	F-Score > 2.45 = elevated risk AUC ≈ **0.69 – 0.74**	Lagged AAER detection · no qualitative/text features · slow signal
Perols 2011 · Auditing JPT	51 fraud + 51 matched controls AAER, size & industry-matched	Six algos compared , LR, SVM, NN, DT, Bayes, kNN	LR & SVM tied at top AUC ≈ **0.65 – 0.72**	Tiny sample (N=102) · matched design overstates performance · won't generalise
Larcker & Zakolyukina 2012 · J. Acc. Research	1,572 conference-call transcripts Q&A segments only	Linguistic deception markers from earnings-call Q&A	CEO answers more predictive than CFO's AUC ≈ **0.65 – 0.69**	Verbal channel only · no fundamentals fusion · small effects · noisy labels
Bao, Ke, Li, Yu, Zhang **2020 · J. Acc. Research**	**28 raw fin. variables** **AAER 1991–2014 · 2.3M firm-yrs**	**RUSBoost , Random Under-Sampling Boosting at scale**	**Beats F-Score & M-Score** **AUC = 0.725 · modern benchmark**	**28 vars only · no text · no event-driven features (8-K, NT 10-K)**

Our gap , No paper combines structured **XBRL fundamentals** with time-aware **EDGAR event signals** (restatements, late filings, auditor changes).

DIAGNOSIS

The original fraud dataset was *statistically meaningless.*

BEFORE

15

POSITIVES

1,274

ROWS

AFTER

86

POSITIVES

2,344

ROWS

✺ *5.7× expansion in positive class*

WHY THE DATA WAS BROKEN

The original Module 2 dataset was built by ***intersecting*** Module 1's rated firms with the fraud label files. The intersection sounded reasonable on paper, but most companies with confirmed AAER violations are ***small-cap or de-listed***, which means they were never publicly rated in the first place. The intersection discarded almost every positive example.

> *Most firms that commit accounting fraud are not the kind of firms Moody's or S&P bothers rating.*

We rebuilt the universe from scratch using the SEC EDGAR XBRL API: ***444 firms***, including every AAER target we could trace, plus a matched clean control. That move alone took positives from ***15 to 86*** and total observations from ***1,274 to 2,344***.

⚠ WHY IT MATTERS

With only 15 positive examples, no classifier could be statistically defensible. A single mislabel would have moved AUC by several points. The audit forced a full data rebuild before any modelling.

DATA

Universe , *444 firms*, 2009 → 2016.

FRAUD SOURCES

99 *firms*

- AAER labels , SEC Accounting & Auditing Enforcement Releases
- SEC litigation releases
- Known misreporting cohort

CONTROL SAMPLE

345 *firms*

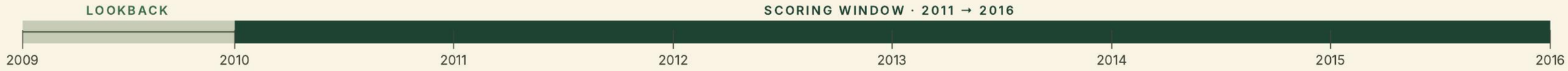
- Top-tickers universe from EDGAR
- No overlap with fraud set
- Stratified by sector + size

FEATURES

60 *features*

- **15** raw XBRL fields
- **41** engineered ratios & flags
- **+4** EDGAR event signals

TEMPORAL COVERAGE



HOW FRAUD FIRMS WERE SOURCED

AAER releases give us SEC-confirmed misstatements with named CIKs. We supplement with scraped litigation releases and a known-misreporting cohort, then dedupe across all three sources.

HOW CONTROLS WERE PICKED

Controls come from EDGAR's top-tickers list, filtered for industry distribution that matches the fraud cohort. We never use a "control" that appears anywhere in the fraud union.

WHY EIGHT YEARS

Scoring runs from 2011 to 2016. The two earlier years (2009 to 2010) are kept only as a ***lookback window*** so three-year rolling features can compute on the first scoring year.

100% public, SEC EDGAR XBRL is open, free, rate-limited, and fully reproducible. *No proprietary data, no vendor feed.*

PREPROCESSING

From raw filings to *model-ready features.*

The discipline is simple: every transformation that *sees data* only ever sees *training rows*. That rule alone closes the four leakage paths we audited later.

01 Pull raw filings from SEC EDGAR

For all 444 companies in our universe, we download the entire history of XBRL-tagged financial figures straight from the SEC website. Numbers like total assets, revenue, net income, debt and cash flow get pulled for every year between 2009 and 2016. We save each company's data locally so we never re-download.

02 Build the features the model will actually read

Raw numbers alone are not very predictive. We turn them into 41 readable features: profitability ratios like return on assets, leverage ratios like debt to assets, year-over-year growth rates, classic accounting red flags (the Beneish ratios), and binary flags for unusual movements like sudden earnings jumps.

03 Add the economic backdrop and sector

A firm is judged differently in a recession than in a boom. We join in eleven macro indicators (interest rates, credit spreads, market volatility, GDP, inflation, unemployment) on fiscal year. We also tag each firm with its sector so the model can compare a tech company to other tech companies, not to a utility.

04 Split the data by company, not by row

Every year of a given company stays in the same bucket, either training or validation. This stops the model from seeing one year of a firm during training and another year during validation, which would let it cheat. Module 1 uses an 80/20 split, Module 2 a 70/30 split.

05 Only learn from training rows when cleaning

Every step that has to learn from data (filling missing values, trimming outliers, rescaling, anomaly detection) is calibrated using only the training rows. The recipe is then frozen and applied to validation rows. This is the single rule that prevented every leakage path we later audited.

06 Hand off to Module 2 with extra fraud signals

Module 1's outputs (predicted rating, residual, importance scores) are bolted onto Module 2's input table. We also merge in four real-time signals from EDGAR filings: did the firm restate? File late? Change auditor? Show unusual insider trading? Final size: 2,344 rows by 131 columns.

ARCHITECTURE

Module 1 *feeds* Module 2.

Read the pipeline left to right. The credit-rating model produces residuals; the fraud model treats those residuals as features. Eight scripts, forty minutes, one seed.

THE BRIDGE

Module 1's ***(predicted minus actual)*** rating gap flows into Module 2 as the ***rating_gap*** feature. Any firm whose books should be rated A but whose agency rating is BBB is a row Module 2 wants to look at.

THE CONTRACT

Join key is ***(ticker, fyear)***. Cardinality grows from 1,845 rated firm-years in Module 1 to 2,344 in Module 2 because unrated fraud firms are now included.

REPRODUCIBILITY

Eight sequential scripts, around forty minutes wall time on a laptop, ***seed = 42*** end to end. Anyone with an internet connection can re-run the whole pipeline from EDGAR.

MODULE 1 · SETUP

Credit rating, *technical spec.*

FIELD	VALUE
TARGET	rating_numeric , ordinal 1–22 (AAA→D)
TRAIN ROWS	**1,475** 80% firm-level
TEST ROWS	**370** 20%
SPLIT	firm-level · seed = 42
FEATURES	**25 total**
, firm fundamentals	8
, macro / yield curve	11
, sector dummies (SIC)	6

Why ordinal, A model that ignores the ordering wastes information. The ordered logit penalises a miss of three notches harder than a miss of one.

FOUR MODELS, ONE TARGET

RIDGE

α = **1.0**
StandardScaler
linear baseline, fast, interpretable

RANDOM FOREST

500 trees, depth 10
min leaf = 5
non-linear, robust to outliers

XGBOOST

300 estimators
depth = 6, lr = 0.05
gradient-boosted benchmark

ORDERED LOGIT

mord.LogisticAT, α = 1.0
structurally ordinal, honours notch distance

Theoretically the right pick is ***Ordered Logit***, since credit ratings are ordinal. But guided by ***Cecchini et al.***, we also bring in Random Forest and XGBoost to capture ***non-linear interactions*** between fundamentals and macro. We average the predictions from all four models, and this ensemble outperforms ordered logit on its own.

MODULE 1 · MACRO

Macro features *carry the model.*

Credit ratings aren't stationary. The ***same balance sheet*** yields different ratings across regimes.

GROUP	FEATURES	WHAT IT READS
YIELD CURVE	dgs10 · dgs3mo · term_spread	Recession expectations priced into bond curve
CREDIT SPREADS	baa_yield · aaa_yield · bbb_oas · baa_aaa_spread	Risk-on / risk-off priced into corporate credit
VOLATILITY	vix	Market stress & uncertainty premium
MACRO CYCLE	unrate · real_gdp_yoy · cpi_yoy	Business cycle, growth, inflation regime

WHY THIS MATTERS

Consider a BBB-rated industrial firm in 2009. The yield curve is steep, credit spreads are blown out, the VIX is over 40. The same firm in 2015 sits in a calm rate regime with tight spreads. ***Identical fundamentals, different macro environment, different rating.***

If we withhold the macro layer, the model overfits to year-specific rating distributions. Hand it the regime, and it learns the relationship between fundamentals ***conditional on the environment***. That is closer to how human credit analysts actually reason.

i. **Counterfactual**
A BBB firm in 2009 is not the same risk as a BBB firm in 2015.

ii. **Generalisation**
Without macro, the model memorises year-specific rating distributions.

iii. **Regime coverage**
Training spans four regimes: recovery, taper tantrum, commodity bust, Brexit shock.

In short, Macro features are not nice-to-haves. They turn the model from *a year-specific lookup* into *a credit-analyst-like reasoner*.

MODULE 1 · RESULTS

RF wins RMSE , *Ordered Logit wins notch accuracy.*

MODEL	RMSE	MAE	ACC@1	ACC@2
Ridge	3.190	2.500	24.9%	49.3%
Random Forest	**3.000 ▲**	**2.230 ▲**	**32.8%**	**55.1%**
XGBoost	3.160	2.400	26.2%	53.5%
Ordered Logit	**3.100**	**2.390**	**41.5% ▲**	**60.9% ▲**

RMSE IN RATING NOTCHES · ACC@K = WITHIN K NOTCHES OF TRUE RATING

HOW TO READ THIS

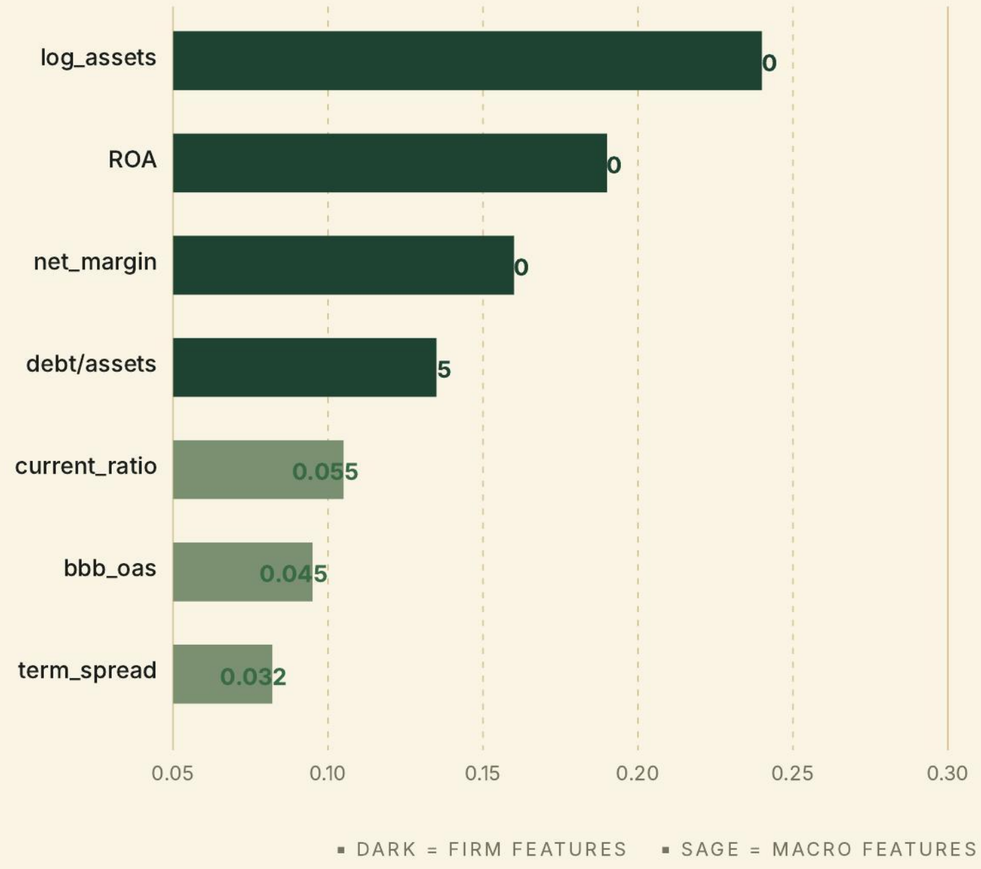
Random Forest has the lowest RMSE (3.0 notches) and lowest MAE (2.2). Ordered Logit has the highest notch accuracy: ***42% within one notch*** and ***61% within two***. The two top models disagree on which loss function matters more, so we keep both as ensemble inputs for Module 2.

Ridge and XGBoost trail. Ridge is a linear baseline; XGBoost overfits on a dataset this small.

Carry-forward, Both winning models pass their predictions, residuals and SHAP attributions into Module 2.

FIG.1 , TOP FEATURES BY RF IMPORTANCE

N = 25



audit triggered

MODULE 2, STAGE 1

Initial results looked great, *too great.*

Module 2 inherits Module 1's outputs as features and adds accounting red flags plus EDGAR event signals. Before the audit, the headline numbers looked impossible.

THE MODULE 2 DATASET

FIELD	VALUE
TARGET	known_fraud_firm (binary)
ROWS	**2,344** firm-years
POSITIVES	86 AAER firm-years (≈ 1 : 26 imbalance)
SPLIT	70 / 30 firm-level, stratified by fraud
FEATURES	**41 total** across four families
from Module 1	rating_pred, rating_gap, SHAP top-k, IG-prob
accounting reds	Beneish M-score + DSRI, GMI, AQI, SGI, LVGI, TATA
EDGAR events	restatement, late filing, auditor change, insider Form 4
deviation flags	earnings jump, accruals spike, post-reversal

STAGE 1 VALIDATION SCORES

MODEL	AUC	F1
PU Learning	**0.977**	**0.809**
Ensemble	0.816	0.516
XGBoost	0.715	0.389

VALIDATION SET · 5-FOLD STRATIFIED · STAGE 1 BUILD

⚠ AUDIT TRIGGERED

AUC 0.977* on a *1 : 26* imbalanced dataset is *not plausible.

Literature on accounting-fraud detection sits at **AUC 0.65 to 0.73**. A jump of 0.30 above that frontier, with the same kinds of features, is a tell. We froze the results and audited all four models for leakage.

MODULE 2, STAGE 2

Four leak paths, *identified and fixed.*

None of these touched labels. That is what made them subtle. Each one let validation rows quietly influence training at a different point in the pipeline.

01 **Leak 1** | IMPUTATION

Median values computed using full dataset
→ Validation rows influenced missing-value filling
FIX | fit medians on train only

02 **Leak 2** | ISOLATION FOREST FIT

Anomaly model learned train + validation structure
→ Validation distribution contaminated training
FIX | fit on training data only

03 **Leak 3** | NORMALISATION

Scaling parameters used full data range
→ Ensemble indirectly saw future distribution
FIX | use train-only min / max

04 **Leak 4** | PU SAMPLING

Validation firms entered unlabeled sampling pool
→ Cross-fold information leakage occurred
FIX | sample from train only

Δ PU LEARNING AUC **0.977** → **0.763** *Δ −0.214*

The gap was the leakage.

MODULE 2 · STAGE 3

Four EDGAR signals with *real fraud lift.*

SIGNAL	SOURCE	FRAUD RATE	CLEAN RATE	LIFT
restatement_within_2yr	**8-K Item 4.02**	**0.150**	**0.021**	**7.26× ▲**
late_filing_flag	**NT 10-K**	**0.150**	**0.029**	**5.13× ▲**
auditor_changes_3yr	8-K Item 4.01	0.260	0.168	1.55×
insider_filing_count	Form 4	12.83	19.21	0.67× (inv.)

FIG.2 , FRAUD LIFT BY SIGNAL

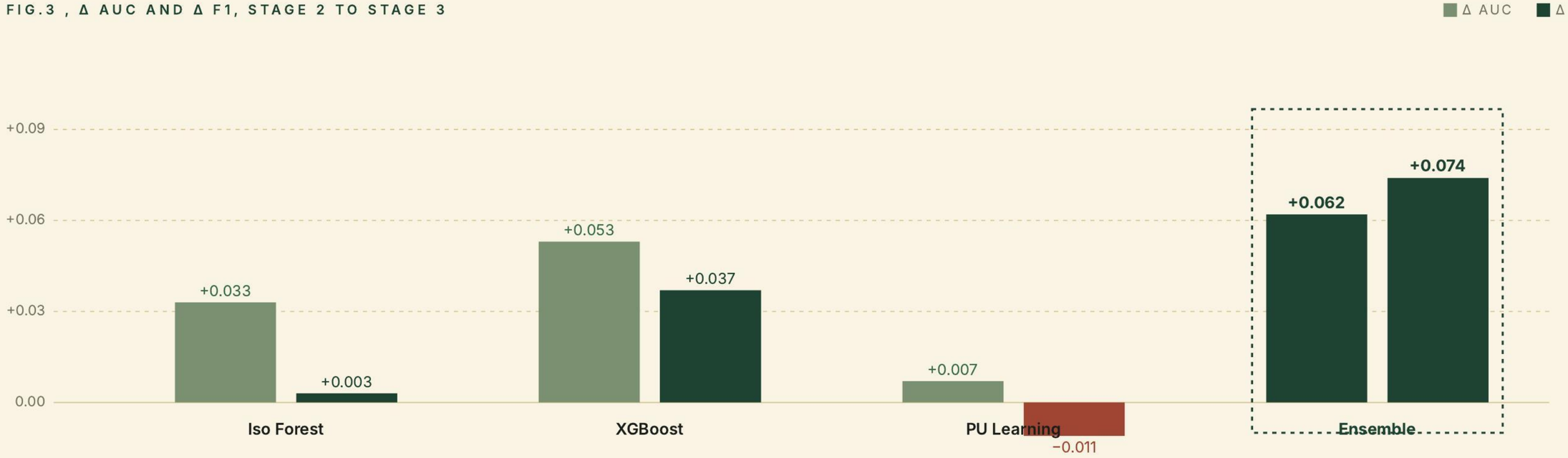
DASHED = NO LIFT (1.0×)

MODULE 2 · STAGE 3 RESULTS

EDGAR signals lift the ensemble *+0.06 AUC, +0.07 F1.*

MODEL	AUC S2	AUC S3	Δ AUC	F1	F2	MCC	RECALL	PRECISION
Isolation Forest	0.480	0.513	**+0.033**	0.294	0.504	0.104	0.956	0.174
XGBoost	0.706	0.759	**+0.053 ▲**	0.400	0.582	0.284	0.838	0.261
PU Learning	0.763	0.770	**+0.007**	0.372	0.555	0.243	0.809	0.245
Ensemble	**0.697**	**0.759**	**+0.062 ▲**	**0.433**	**0.601**	**0.331**	**0.809**	**0.296**

FIG.3 , Δ AUC AND Δ F1, STAGE 2 TO STAGE 3



MODULE 2, THE PRECISION QUESTION

Stage 3 was *recall-strong*, precision-weak.

We catch four out of five frauds, but two of every three alerts are false positives. F1 alone hides this tension. F2 and MCC make it visible.

METRIC	VALUE	WHAT IT MEASURES	WHY WE CARE
Recall	**0.809**	**Share of real frauds the model flagged**	**Four out of five caught**
Precision ⚠	0.296	Share of flags that turn out to be real fraud	One in three is genuine
F1	0.433	Harmonic mean of precision and recall	Standard summary metric
F2 ★	**0.601**	**Recall-weighted F-score (β = 2)**	**Matches the deployment cost function**
MCC ★	**0.331**	**Balanced correlation across all 4 confusion cells**	**Robust to the 1 : 26 class imbalance**
AUC	0.759	Probability the model ranks a fraud above a clean firm	Vs literature band 0.65 to 0.73

Why F2. A missed fraud costs an auditor roughly 100 times a false alarm. F2 weights recall four times more than precision, which is what the cost function actually looks like in practice.

Why MCC. With a 1 : 26 imbalance, accuracy and F1 can both inflate even when the model is mediocre. MCC penalises false alarms and false misses equally and only rewards joint TP and TN.

For auditor review, Ideal target is *precision above 0.60 at non-trivial recall.* The next slide walks through the four experiments we ran to try to get there.

MODULE 2, FOUR PRECISION EXPERIMENTS

What we tried, *and what it taught us.*

Stage 3 left us with strong recall and weak precision. We ran four follow-ups to push precision upward. One worked. The other three told us something more useful: the bottleneck is feature quality, not the loss function.

01 Neural network with weighted BCE and focal loss

marginal

We trained four MLP variants (balanced, conservative, small, deep) using sample weights tuned to the 1:12 imbalance, plus a focal-loss variant. **Best variant:** mlp_weighted_bce_conservative reached $F_{0.5} \approx 0.41$. The result moved the needle a single point and depended heavily on the random seed. Verdict: not worth the complexity over the existing logistic ensemble.

02 Hard-negative mining on Stage-1 false positives

negative result

The idea: take firms the Stage-1 model flagged as fraud but that turned out clean, train a second-stage classifier only on those hard negatives versus real positives. **Best variant:** extra-trees stage-2 reached $F_{0.5} \approx 0.39$. Verdict: the hard negatives do not separate from real fraud on the features available, so the second stage adds no signal.

03 Cost-sensitive XGBoost with FP penalty grid

pareto point

Penalise false positives harder by scanning a grid over *fp_cost* and *pos_weight*. **Best variant:** raw_cost_xgb_fp1_pos0.75 hits *precision = 0.60 at recall = 0.31*, with $F_{0.5} = 0.53$. Verdict: a genuine Pareto trade. You buy precision by giving up recall. Useful if the deployment context demands precision near 0.60.

04 Refinitiv vendor features merged into the dataset

negative result

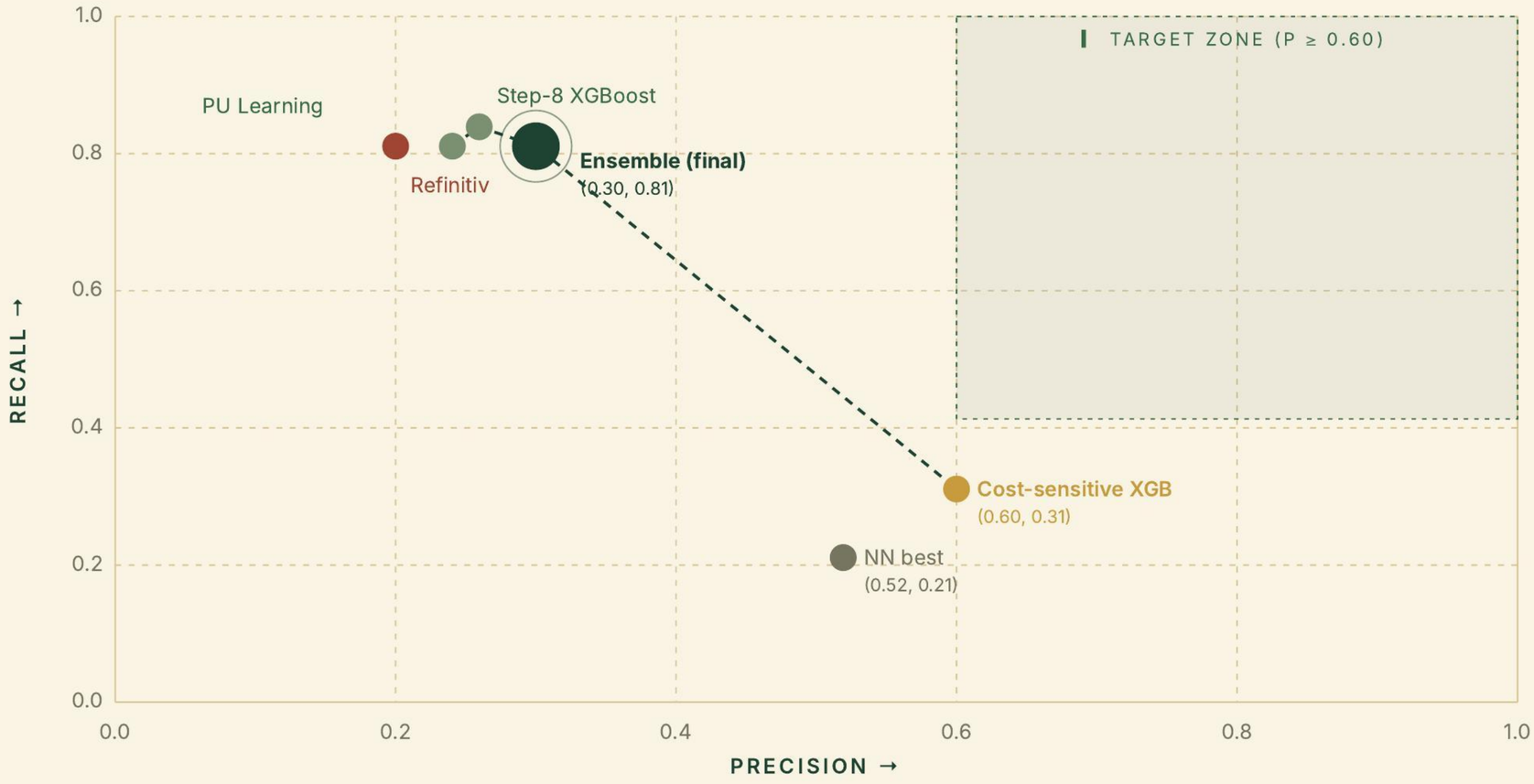
Plug in external vendor signals (ESG scores, governance metrics, analyst revisions). Result: *AUC 0.76 to 0.62*, AP *0.33 to 0.25*. Verdict: an honest negative. Coverage is sparse on small-cap fraud firms, and most of what Refinitiv encodes is already implicit in our XBRL plus EDGAR feature set. More data is not always better data.

MODULE 2 · PRECISION–RECALL FRONTIER

We are *feature-bounded*, not algorithm-bounded.

FIG.4 , PRECISION–RECALL FRONTIER (VALIDATION)

DASHED = PARETO FRONTIER



LEGEND

- Ensemble (final)
- Stage-3 baselines
- Cost-sensitive XGB
- Neural Network
- Refinitiv variants

READING

All four experiments stayed *on or below* the existing frontier. The bottleneck is **feature quality**, not loss function.

FINAL MODEL

Ensemble meta-learner, *deployed.*

A logistic regression sitting on top of three other models. Eleven normalised inputs go in, one fraud probability comes out. The simplest possible stack that beats every individual track.

THE ELEVEN INPUT SIGNALS

- *iso_norm*
- *xgb_score*
- *pu_score*
- *prob_boundary*
- *restatement_2yr*
- *auditor_changes_3yr*
- *beneish_m_norm*
- *suspicion_norm*
- *gap_consensus_norm*
- *beneish_flag*
- *late_filing*

> *Three model scores plus engineered signals plus EDGAR event flags. The meta-learner just decides how much weight to give each one.*

VALIDATION METRICS

AUC		0.759
AP	average precision	0.330
PRECISION		0.296
RECALL		0.809
F1 ★	best of 4 tracks	0.433
F2 ★	deployment metric	0.601
MCC ★	imbalance-robust	0.331

BENCHMARKS

Where our model sits *in the literature.*

PAPER	YEAR	METHOD	AUC RANGE	Δ VS OURS
Beneish , M-Score	1999	Probit, 8 ratios	0.65 – 0.70	**+0.06 – +0.11**
Cecchini et al. , Mgmt Sci	2010	SVM, financial kernel	0.70 – 0.73	**+0.03 – +0.06**
Dechow et al. , F-Score	2011	Logistic regression	0.69 – 0.74	**+0.02 – +0.07**
Perols , Auditing JPT	2011	LR / SVM / NN / DT	0.65 – 0.72	**+0.04 – +0.11**
Larcker & Zakolyukina , JAR	2012	Linguistic (earnings calls)	0.65 – 0.69	**+0.07 – +0.11**
Bao et al. , JAR	2020	RUSBoost, raw fin.	0.725	**+0.034**
THIS WORK	**2026**	**Ensemble + EDGAR signals**	**0.759**	„

FIG.5 , AUC VS PUBLISHED BENCHMARKS

SORTED ASCENDING

⚠ INDICATIVE, NOT APPLES-TO-APPLES · PERIODS, AAER VINTAGES & VALIDATION PROTOCOLS DIFFER.

METHODOLOGY

How we made sure we weren't *fooling ourselves.*

The first time we ran the model, the AUC came out at 0.977. That was a red flag. We worked backwards through the pipeline and found seven small habits we had picked up that were letting the validation data secretly help the training data. We fixed each one and re-ran.

✓ **Split by company, not by year**
If 2014 of a firm trained the model, 2015 of the same firm cannot show up in validation.

✓ **Fill in missing numbers from training only**
Median values to plug holes are computed using training rows. Validation rows do not get to influence the fill.

✓ **Teach the anomaly detector on training rows**
The Isolation Forest only learns what normal looks like from training data, then scores the rest.

✓ **Rescale signals using training ranges**
When the ensemble compresses signals to [0, 1], the min and max come from training, never recomputed later.

✓ **Draw unlabeled samples from training only**
When PU learning needs unlabeled firms as pseudo-negatives, we only pull them from the training pool.

✓ **Fit the scaler once and reuse it**
StandardScaler is fit on the training subset and then applied unchanged everywhere else.

✓ **No looking into the future**
EDGAR event signals are taken from the window 30 to 730 days after the firm's year-end, so the model never reads news that did not yet exist.

✓ **One random seed, used everywhere**
Seed 42 for every split, sample, and shuffle. Anyone re-running the code will see the same numbers.

What this cost us, After all seven fixes, our AUC dropped from *0.977 to 0.763*. We were not happy about it, but the new number is one we actually believe.

APPLICATIONS

Where this model operates *in practice.*

01 | CREDIT RATING AGENCIES

Moody's · S&P · Fitch

Augment human analyst judgment with a probabilistic prior on rating notch (Module 1) and a fraud overlay on rated issuers.

USE CASE → **EARLY FLAG BEFORE DOWNGRADE COMMITTEE**

02 | BANK CREDIT RISK

Portfolio screening

Score the full corporate-loan book quarterly. Obligors whose rating-gap or restatement risk exceeds threshold raised to relationship manager.

USE CASE → **BASEL-ALIGNED EARLY-WARNING**

03 | AUDIT FIRMS (BIG 4)

Audit-risk assessment

Pre-engagement risk scoring on new clients. Allocates audit hours proportional to ensemble score.

USE CASE → **PCAOB ENGAGEMENT PLANNING**

04 | REGULATORS

SEC · PCAOB · FINRA

Prioritise enforcement & inspection workload. Top-50 precision **0.38** = **19 / 50** reviewed firms genuinely problematic, vs ~4% base rate.

USE CASE → **AAER CANDIDATE TRIAGE**

05 | INVESTMENT MANAGEMENT

Long / short alpha

Short-selling alpha , flagged firms underperform on drift to restatement / downgrade. ESG / governance screen for institutional mandates.

USE CASE → **FACTOR OVERLAY ON QUANT FUNDAMENTALS**

06 | INSURANCE & COUNTERPARTY

D&O · E&O · credit ins.

Underwriting input for Directors & Officers, financial-statement misrepresentation, and corporate credit insurance.

USE CASE → **PREMIUM CALIBRATION & EXCLUSIONS**

A ranking tool, not a verdict. The model is designed for **human-in-the-loop review**; thresholds tuned per vertical.

REFLECTION

What we learned · what remains.

LESSONS

- Question results that look *too good*.
- Leakage hides in *imputation, scaling, sampling* , not just labels.
- *Feature quality > algorithm choice* in the tail.
- Class imbalance is not always a *"problem to fix"*.
- Honest baselines tell *better stories* than tuned ones.

LIMITATIONS / NEXT STEPS

- Forward-time holdout (train ≤ 2014 / test ≥ 2015).
- Per-sector ensembles to reduce within-industry variance.
- NLP on 10-K MD&A , *FinBERT* embeddings.
- Probability calibration for cost-aware thresholds.
- Larger fraud pool via *non-AAER* label sources.

thank you.

Ready for *review.*

$ ./pipeline.sh --seed 42 --validate // status: complete

PIPELINE
8 scripts
~40 min end-to-end

DATA
SEC EDGAR
100% public XBRL

REPRODUCIBLE
seed = 42
throughout

FINAL
AUC 0.759 · F1 0.433
ensemble meta-learner

AUTHORS

Avneet Kaur · Hunnar Khurana · Renaya Gupta