ASSESSING MACHINE-LEARNING MODEL ROBUSTNESS

PREPARED BY: YVES ASALI & LUCAS DOAN

Agenda

McGill

LITERATURE REVIEW

Assessing Robustness of Machine Learning Models Using Covariate Perturbations

By Arun Prakash, Anwesha Bhattacharyya, Joel Vaughan, and Vijayan N. Nair (Wells Fargo, 2024)

Focus: Covariate perturbation techniques and diagnostics for tabular model robustness.

Towards Evaluating the Robustness of Neural Networks

By Nicholas Carlini & David Wagner (2017)

Focus: Foundational adversarial attacks (L_2, L_0, L_∞) and critique of defenses like defensive distillation.

TabularBench: Benchmarking Adversarial Robustness for Tabular Deep Learning in Real-world Use-cases

By Tommaso Simonetto, Soufiane Ghamizi, Maxime Cordy (2024)

Focus: Realistic adversarial benchmarks and architecture/training strategies for structured data.

An Empirical Study of Accuracy, Fairness, Explainability, Distributional Robustness, and Adversarial Robustness

By Moninder Singh et al. (2021)

Focus: Multi-dimensional model evaluation across common ML architectures and datasets.

Machine Learning Robustness: A Primer

By Houssem Ben Braiek & Foutse Khomh (2024)

Focus: A comprehensive, conceptual and metric-driven framework for robustness assessment and enhancement.

DATA OVERVIEW

Data Cleaning Strategy Overview

Purpose of Cleaning:

- Eliminate noise and reduce overfitting.
- Prepare high-quality inputs aligned with TabNet's and XGBoost architecture.
- Handle missing data in a principled, context-aware manner.

Core Steps:

- **Dropped irrelevant or low-quality features**:
  - Real estate details with >60% missing values (e.g., apartment size, number of entrances).
  - Identifier columns (e.g., previous application IDs).
  - Flags and indicators with limited predictive value (e.g., document flags).
- **Imputed missing values thoughtfully** based on domain context:
  - **Zeros** for credit usage, timelines, and bureau features indicating no activity.
  - **Medians** for predictive numeric variables.
  - **Modes** for count-based features.
  - **'Unknown'** for missing categorical values.

Outcome:

- Clean, lean dataset with good predictive power.
- Aligned preprocessing with Models design.

DATA OVERVIEW

Imputation Strategy by Feature Type

Zero Imputation:

- For credit usage or delinquency (e.g., credit limit, drawing amounts, days past due) — assumed absence of credit activity.
- For bureau requests and payment timelines — interpreted missing values as no request or event.

Median Imputation:

- For predictive risk scores (e.g., external credit scores from third-party sources) — preserves overall distribution without skewing.
- For continuous behavioral indicators (e.g., days since last phone change, early repayment flag).

Mode Imputation:

- For count-based variables like number of family members — using the most frequent value to maintain consistency.

Categorical Imputation:

- For occupation type, housing type, and similar — missing values replaced with `'Unknown'` to preserve category structure without adding artificial bias.

DATA OVERVIEW

Feature Reduction and Final Dataset Integrity

Dropped Features:

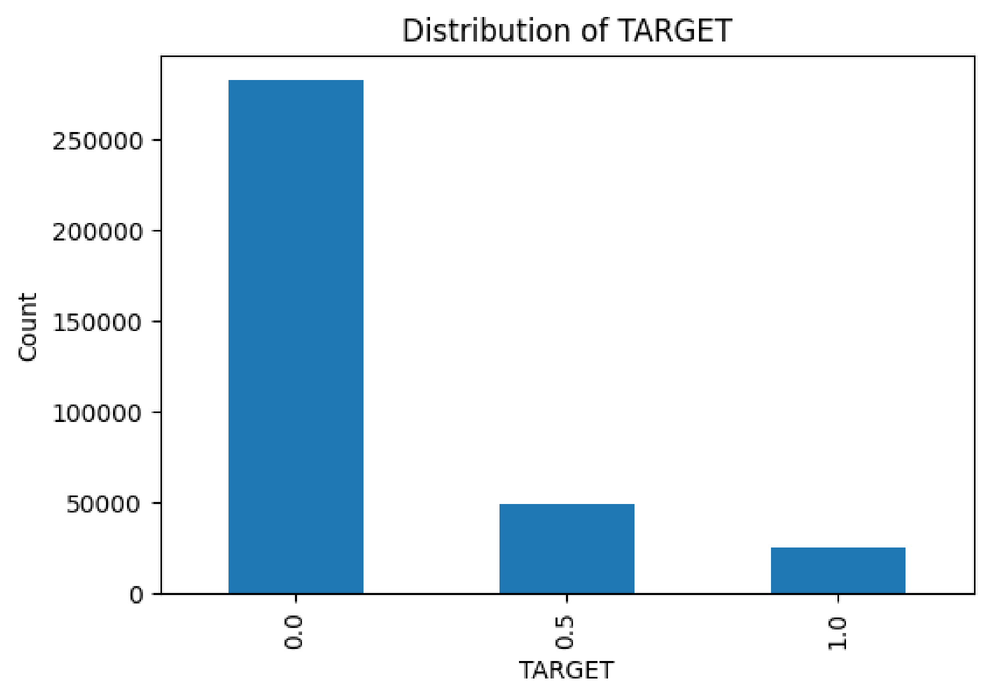
- **High-Missing Real Estate Fields**: Land area, number of elevators, basement area — unreliable due to data sparsity.
- **Identifiers**: Internal linking IDs offered no predictive value.
- **Document Flags**: Noisy indicators that added minimal incremental value.

Other Quality Checks:

- **No constant (zero-variance) features** were found — all retained columns had variability.
- **No highly correlated pairs** above 0.9 — ensured model robustness and interpretability.

Outcome:

- Dimensionality reduced without loss of signal.
- The final dataset is reliable, interpretable, and ready for modeling across different algorithms.



EVALUATION TECHNIQUE- COVARIATE PERTURBATION

This method systematically perturbs the input data to evaluate the **robustness of a trained classifier** under realistic data degradation scenarios. By applying controlled noise or masking to test features, we assess how sensitive the model is to small but meaningful variations. This helps identify **failure modes and generalization weaknesses**.

•**Gaussian Noise Perturbation**
Adds random noise to each feature to simulate measurement errors or sensor noise.
→ *Tests model tolerance to minor, continuous fluctuations in feature values.*

•**Feature Shift Perturbation**
Scales all features by a constant factor (e.g. +10%).
→ *Simulates covariate shift from data drift, environmental change, or calibration errors.*

•**Random Mask Perturbation**
Randomly zeroes out features with a given probability.
→ *Models missing or unreliable data (e.g. sensor dropout, partial records).*

EVALUATION TECHNIQUE- THREE PRECISE ATTACKS

L_0-Norm Constrained Attacks ($\|\tilde{x} - x\|_0 \leq k$): Only a small number (k or fewer) of input features can be changed; the rest must remain untouched.

1. Gradient-Based L_0: This method calculates how much each input feature influences the model's prediction using gradients, then changes the k most influential features in a way that increases prediction error.

2. Loss-Sensitive L_0: Instead of relying on gradients, this method adds small test noises to each feature to see which ones most increase the model's loss, then selects and perturbs the k features with the largest impact.

L_2-Norm Constrained Attacks ($\|\tilde{x} - x\|_2 \leq \varepsilon$): Changes can be spread across many features, but the total magnitude (energy) of all changes must stay below ε.

3. FGSM-L2: This fast one-step attack moves the input slightly in the direction that most increases model error, while scaling the movement so that the total change stays within a defined L_2 distance.

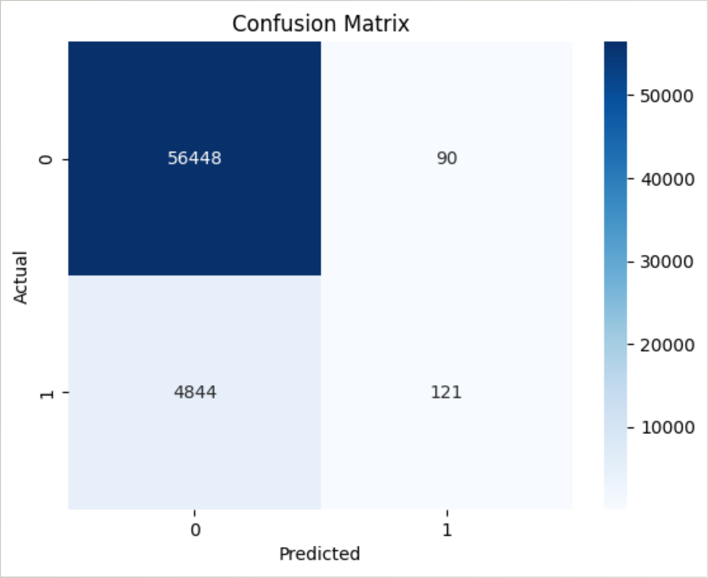
4. PGD-L2: This stronger, iterative version of FGSM applies several small steps to gradually increase the model's loss, projecting the result back to ensure the total change stays within the allowed L_2 distance after each step.

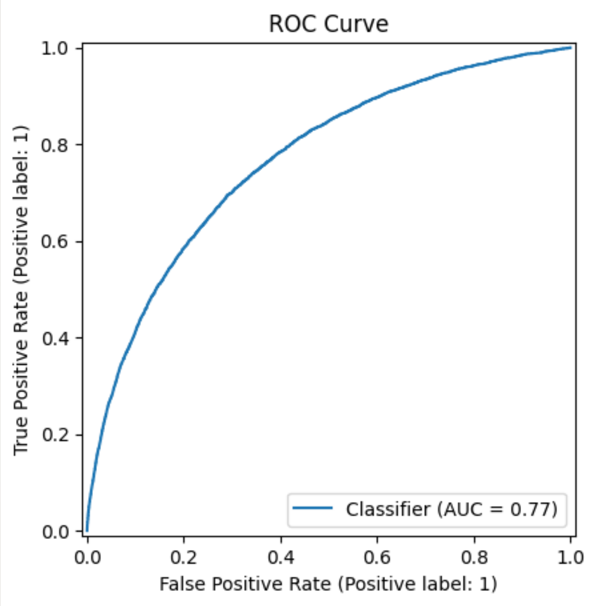
L_∞-Norm Constrained Attacks ($\|\tilde{x} - x\|_\infty \leq \varepsilon$): Each individual feature is allowed to change only a little (by no more than ε), but many features can be changed simultaneously.

5. FGSM-L_∞: This simple and fast attack changes all features at once in the direction that increases the model's loss, limiting each individual change to be within ε.

6. PGD-L_∞: This is a stronger, multi-step version of FGSM that applies repeated small changes to increase loss; while making sure each feature stays within the maximum allowed change after every step.

EXTREME GRADIENT BOOSTING - XGBR





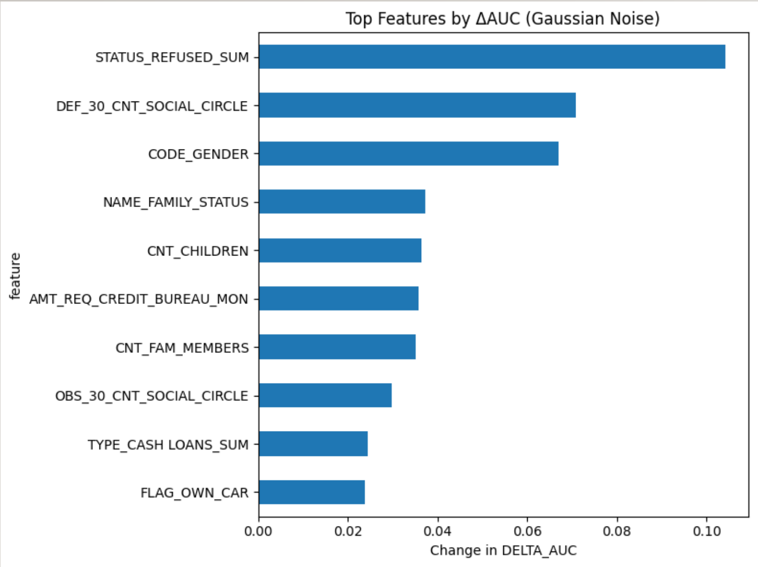
Test AUC: 0.7676 → Fair model separation ability.
High Accuracy (92%) but misleading due to imbalance.
Class 1 Recall: Only 2% → Model fails to detect minority class.
F1 Score (Class 1): 0.06 → Extremely weak performance on Class 1.
Strong performance on Class 0 (F1 = 0.96) shows model is skewed toward the majority.

Label	Precision	Recall	F1-Score	Support
Class 0	0.92	1.00	0.96	56,538
Class 1	0.49	0.06	0.11	4,965
Accuracy			**0.92**	61,503
Macro Avg	0.71	0.53	0.53	61,503
Weighted Avg	0.89	0.92	0.89	61,503

Takeaway:
The model achieves high overall accuracy (92%) but **fails to detect most housing credit defaults**, with only **6% recall for defaulters.**

McGill

COVARIATE PERTURBATION RESULT



Gaussian Noise (5% noise, 10% mask)

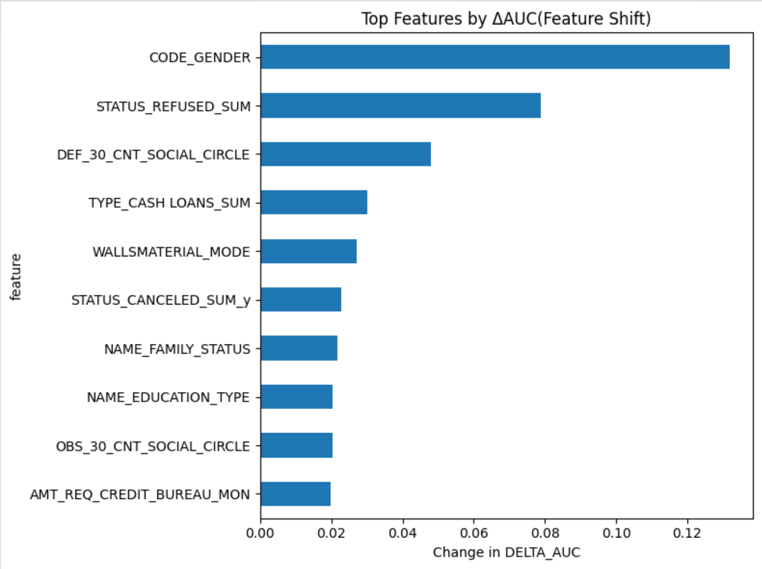
•**Class 0 remains highly stable**: Precision = 0.93, Recall = 0.95 → excellent detection of non-defaults despite noise.

•**Minimal prediction flips**: Only **6.2%** of predictions changed.

•Class 1 remains weak (Recall = 0.15), but that's expected due to class imbalance.

•**AUC-ROC drops to 0.65**, showing the model's separation boundary is moderately sensitive to small continuous noise.

Model shows **good resilience for Class 0** with minor performance degradation, but is still weak at detecting rare defaults.



Feature Shift (+10% scaling, 10% masking)

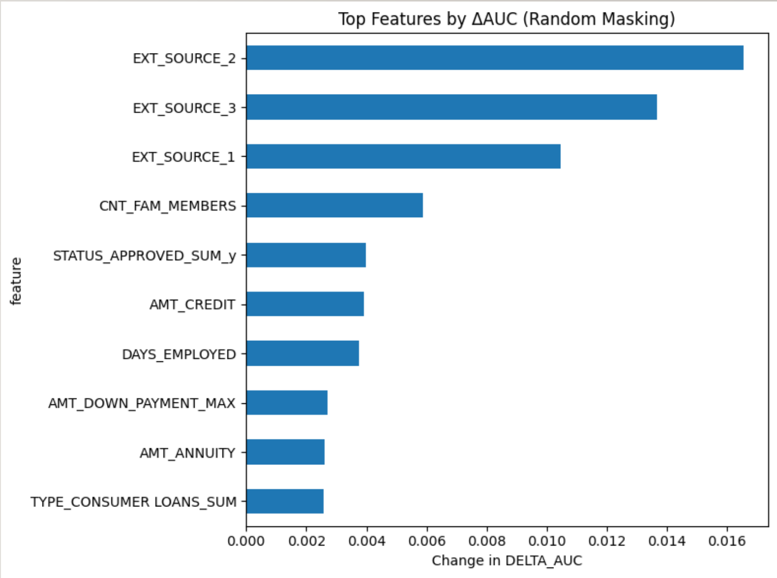
•**Class 0 near-perfect recall** (1.00) — model overconfidently classifies everything as non-default.

•**Only 0.96% predictions changed**, which shows **extreme stability.**

•**Default detection collapsed**: Class 1 recall = 0.03 → defaults are almost entirely ignored.

•AUC-ROC = 0.75 → suggests the model still "ranks" well but **threshold-based performance is poor**.

Model is extremely brittle toward shift when it comes to Class 1 — it holds on to Class 0 predictions too tightly.



Random Masking (15% features missing)

•**Class 0 predictions hold strong**: Recall = 0.98, F1 = 0.95

•Class 1 recall = 0.09, slightly better than feature shift but still poor.

•**2.57% of predictions changed**, which shows **good tolerance to missing data**.

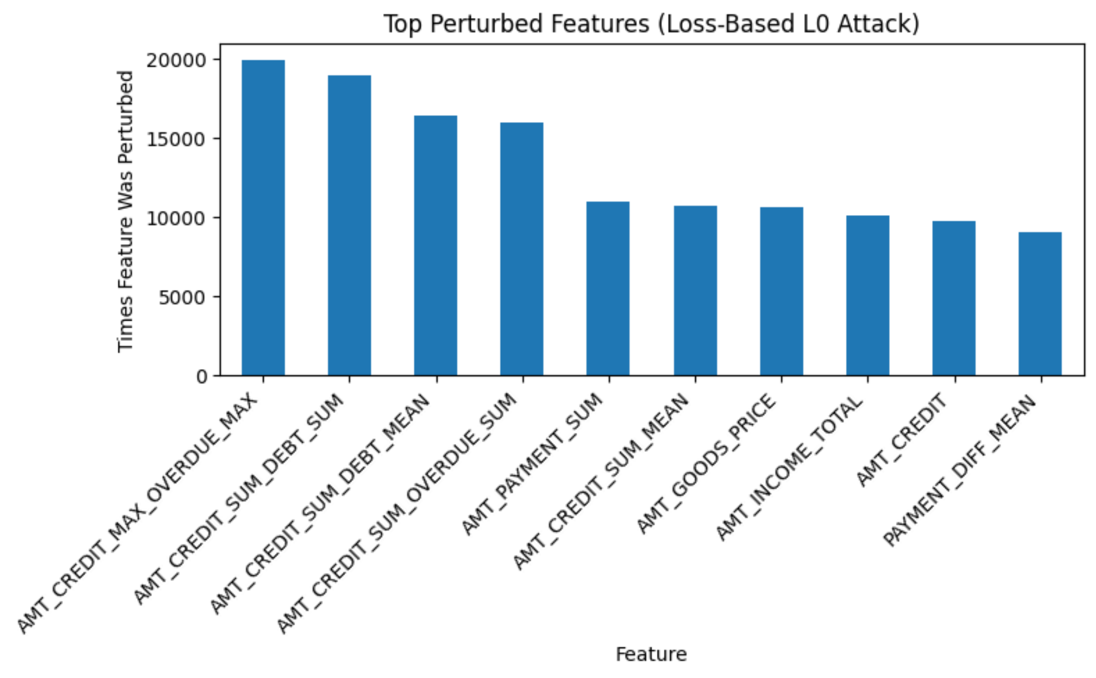
•AUC-ROC = 0.696 → moderate impact on separability.

Model handles missing data well for non-defaults, making it robust in real-world scenarios where input features may be incomplete. Still underperforms on Class 1 due to imbalance, not just data noise.

L0 Attack Result

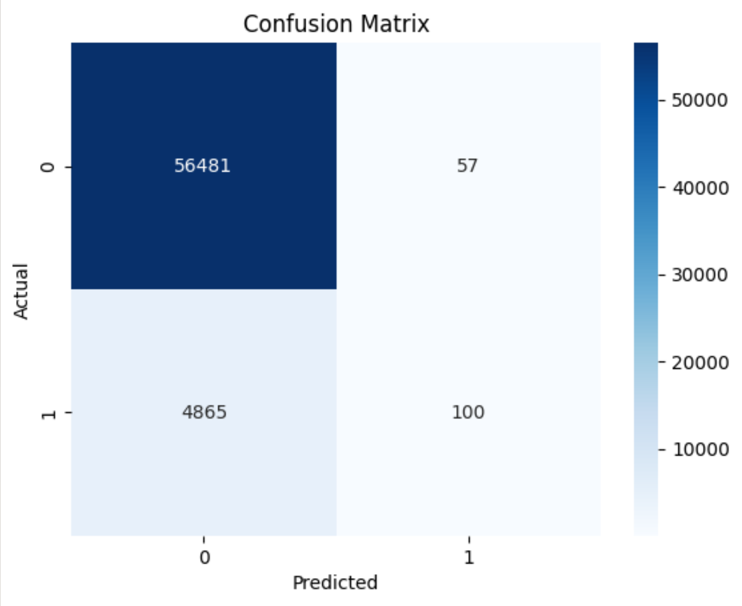
Loss-Based L_0 Attack.
•**AUC-ROC**: 0.4176
→ **Significant drop** from baseline 0.7676, indicating **severe loss in discriminative power**. The model can barely distinguish between classes post-attack.
•**Accuracy**: 0.8787
→ This metric appears deceptively high due to **extreme class imbalance**. Since Class 0 dominates the dataset, the model achieves high accuracy by defaulting to majority predictions—even if it's nearly useless for Class 1.
•**Class 1 Performance**:

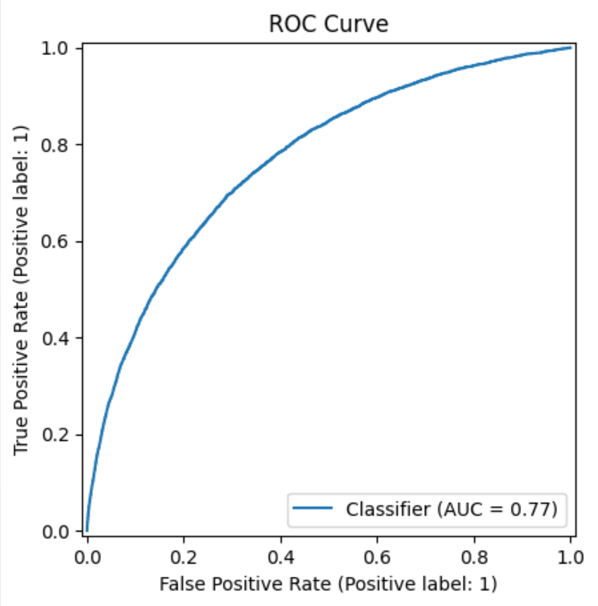
- **F1-score**: 0.0281
- **Recall**: 0.02
  → These values are **catastrophically low**. The model is failing to capture positive (Class 1) cases under the L_0 attack, which flips only a few key features.



Interpretation: These features are likely contributing most to the model's decision boundaries. Their high perturbation count implies potential over-reliance or lack of redundancy—particularly for credit-related and payment-history variables.

TABNET



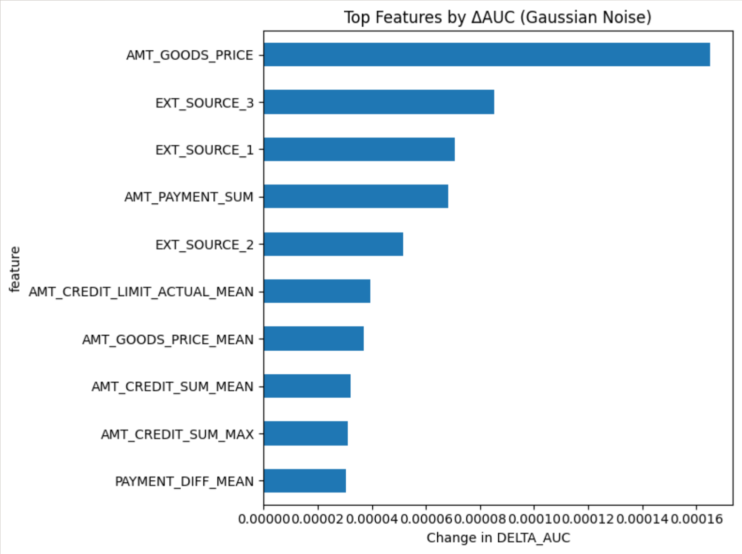


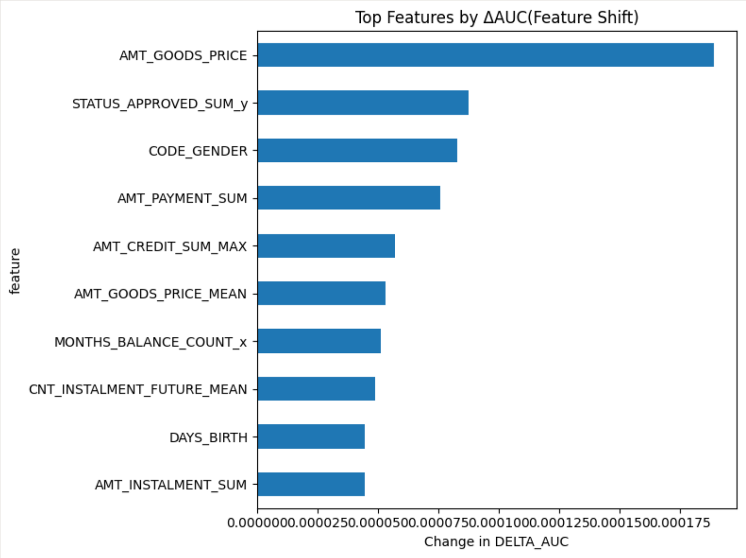
Test AUC: 0.7678 → Fair model separation ability.
High Accuracy (91.98%) but misleading due to imbalance.
Class 1 Recall: Only 2% → Model fails to detect minority class.
F1 Score (Class 1): 0.05 → Extremely weak performance on positives.
Strong performance on Class 0 (F1 = 0.96) shows model is skewed toward the majority.

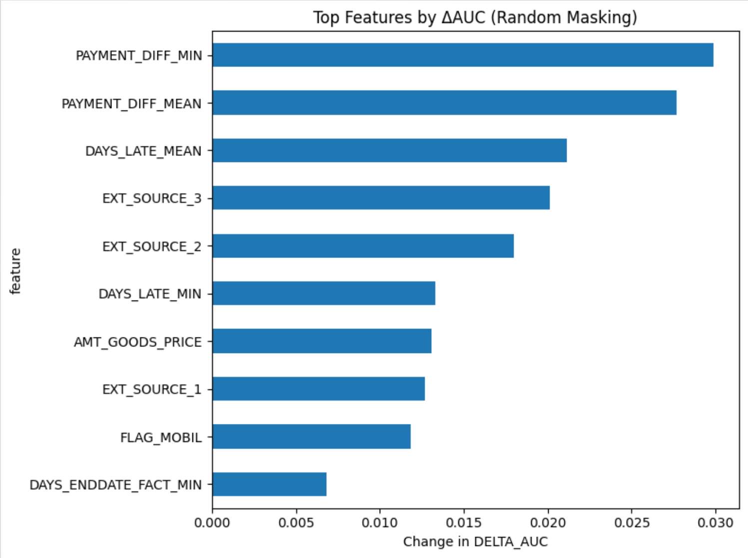
Label	Precision	Recall	F1-Score	Support
Class 0	0.92	1.00	0.96	56,538
Class 1	0.57	0.02	0.05	4,965
Accuracy			**0.92**	61,503
Macro Avg	0.75	0.51	0.50	61,503
Weighted Avg	0.89	0.92	0.88	61,503

Takeaway:
The model achieves high overall accuracy (92%) but **fails to detect most housing credit defaults**, with only **5% recall for defaulters.**

COVARIATE PERTURBATION- RESULT







Gaussian Noise Perturbation

•**AUC-ROC**: 0.5895 (↓ from 0.7729)

•**F1 (Class 1)**: 0.09, **Recall**: 0.06

•**Prediction changes**: 3.67%

•**Top sensitive features**: AMT_GOODS_PRICE, EXT_SOURCE_3, AMT_PAYMENT_SUM

•**Summary**: Mild noise caused a sharp AUC drop and Class 1 degradation. Model is fragile to small, realistic perturbations in key credit/payment fields.

Feature Shift Perturbation

•**AUC-ROC**: 0.7293

•**F1 (Class 1)**: 0.02, **Recall**: 0.01

•**Prediction changes**: 0.24%

•**Top sensitive features**: AMT_GOODS_PRICE, STATUS_APPROVED_SUM_y, CODE_GENDER

•**Summary**: Minor distribution shifts barely change outputs but cripple minority class detection. Model is confidently wrong—dangerous in real-world drift.

Random Mask Perturbation

•**AUC-ROC**: 0.6519

•**F1 (Class 1)**: 0.11, **Recall**: 0.07

•**Prediction changes**: 2.74%

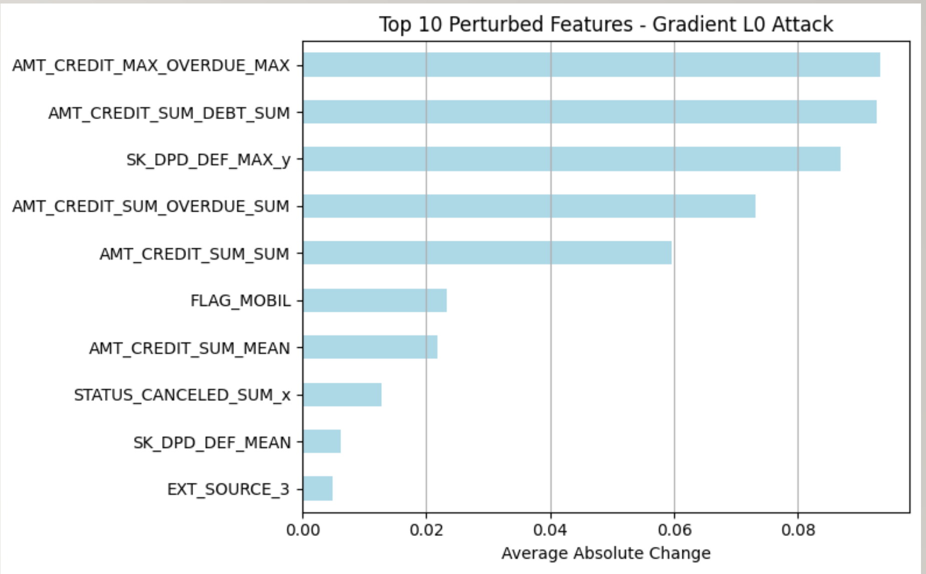
•**Top sensitive features**: AMT_PAYMENT_SUM, AMT_CREDIT_SUM_MEAN, PAYMENT_DIFF_MEAN

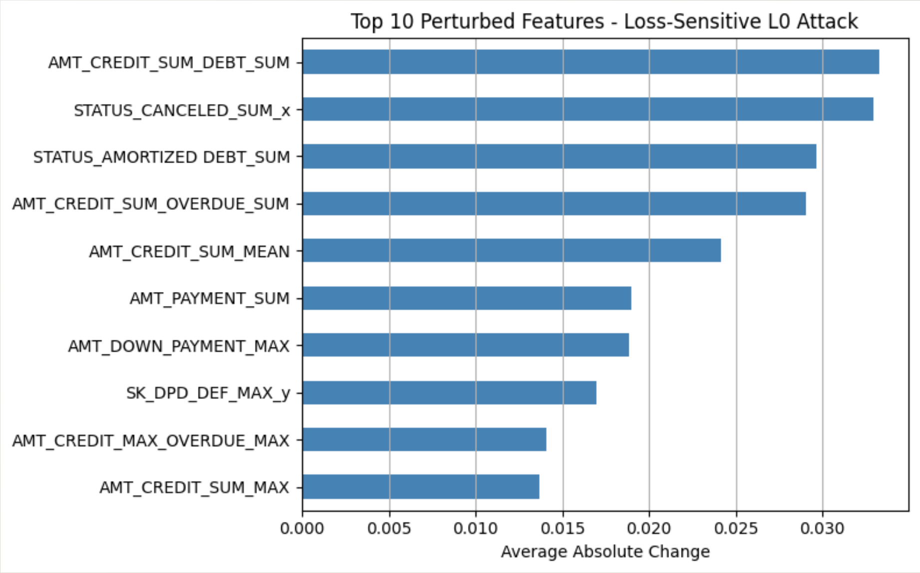
•**Summary**: Missing data breaks the model moderately. Slightly more stable than Gaussian, but still weak fallback logic for partial input loss.

L0 RESULTS

Gradient-Based L_0 Attack
•**Class 0**: Precision = 0.91, Recall = 0.93, F1 = 0.92
•**Class 1**: Precision = 0.01, Recall = 0.01, F1 = 0.01
•**AUC = 0.0668**, **Accuracy = 0.8597**, **Overall F1 = 0.0117**
•**Top perturbed features**: AMT_CREDIT_MAX_OVERDUE_MAX, AMT_CREDIT_SUM_DEBT_SUM, SK_DPD_DEF_MAX_y, AMT_CREDIT_SUM_OVERDUE_SUM, AMT_CREDIT_SUM_SUM, FLAG_MOBIL, AMT_CREDIT_SUM_MEAN, STATUS_CANCELED_SUM_x.
Insight: Despite the high accuracy and Class 0 dominance, sparse perturbations targeting credit and delinquency indicators reveal fragility — a few key changes are enough to collapse Class 1 detection and degrade Class 0 confidence.

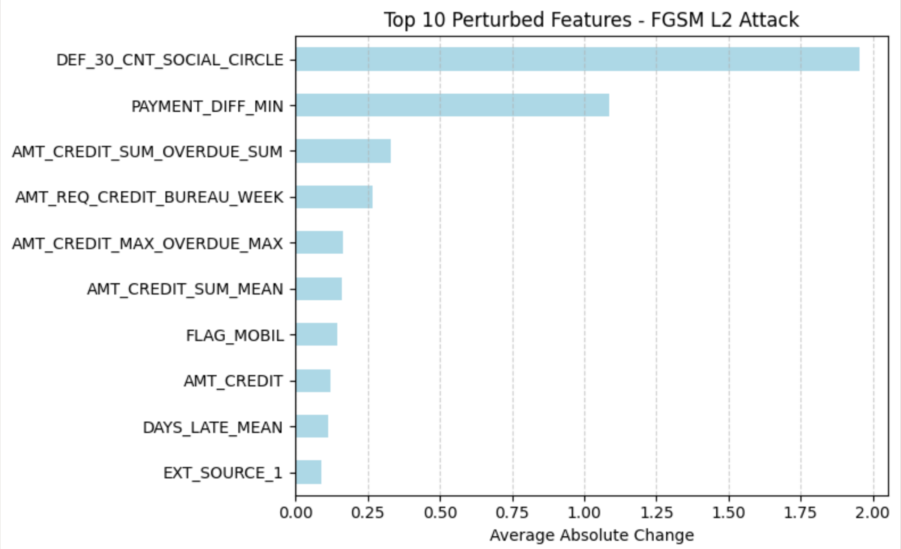
Loss-Sensitive L_0 Attack
•**Class 0**: Precision = 0.92, Recall = 0.94, F1 = 0.93
•**Class 1**: Precision = 0.05, Recall = 0.03, F1 = 0.04
•**AUC = 0.4593**, **Accuracy = 0.8645**, **Overall F1 = 0.0390**
•**Top perturbed features**: AMT_CREDIT_SUM_DEBT_SUM, STATUS_CANCELED_SUM_x, STATUS_AMORTIZED_DEBT_SUM, AMT_CREDIT_SUM_OVERDUE_SUM, AMT_CREDIT_SUM_MEAN, AMT_PAYMENT_SUM, AMT_DOWN_PAYMENT_MAX.
•**Insight**: This attack more selectively targets features that drive the model's loss — still mostly related to credit burden — leading to a sharper impact on Class 1 while barely denting Class 0. This imbalance highlights a core issue: the model maintains Class 0 recall but does so at the cost of completely ignoring minority risks.



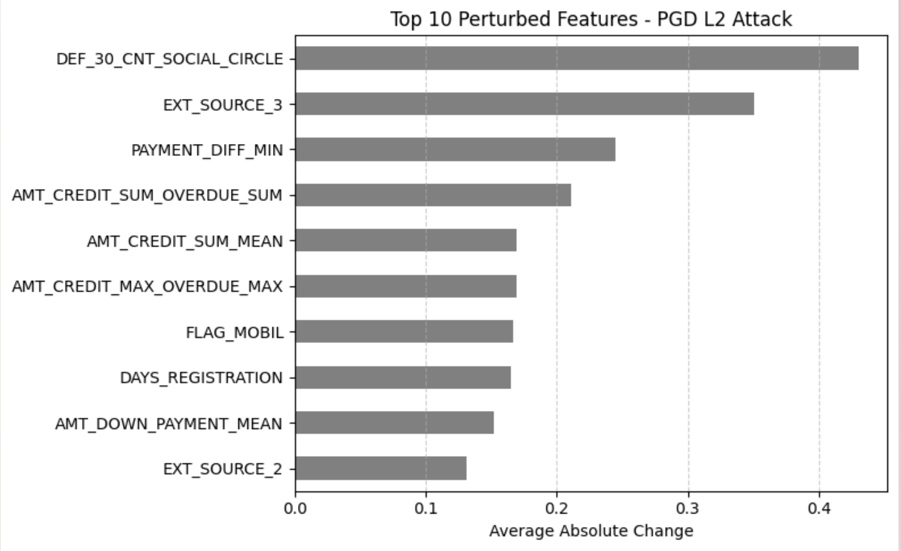


L2 Results

FGSM L_2 Attack
•**Class 0**: Precision = 0.93, Recall = 0.77, F1 = 0.84
•**Class 1**: Precision = 0.12, Recall = 0.37, F1 = 0.19
•**AUC = 0.6071**, **Accuracy = 0.7386**, **Overall F1 = 0.1853**
•**Top perturbed features**: AMT_CREDIT_MAX_OVERDUE_MAX, SK_DPD_DEF_MAX_y, AMT_CREDIT_SUM_OVERDUE_SUM, AMT_CREDIT_SUM_DEBT_SUM, AMT_CREDIT_SUM_SUM.
•**Insight**: FGSM reveals that even light L2 perturbations can erode Class 0 recall by over 20%, showing the model's overreliance on overdue and debt-related features.



PGD L_2 Attack
•**Class 0**: Precision = 0.74, Recall = 0.22, F1 = 0.34
•**Class 1**: Precision = 0.02, Recall = 0.15, F1 = 0.03
•**AUC = 0.1383**, **Accuracy = 0.2122**, **Overall F1 = 0.0296**
•**Top perturbed features**: STATUS_CANCELED_SUM_x, EXT_SOURCE_2, AMT_CREDIT_MEAN, EXT_SOURCE_3, AMT_CREDIT_MAX_OVERDUE_MAX, SK_DPD_DEF_MAX_y, STATUS_1_SUM_SUM, AMT_CREDIT_SUM_DEBT_SUM, AMT_CREDIT_SUM_OVERDUE.
•**Insight**: PGD breaks the model completely, targeting risk source scores and long-term credit indicators. Class 0, which dominates the data, loses most of its recall — showing deep instability under iterative attacks.



L∞ Result

FGSM L∞ Attack

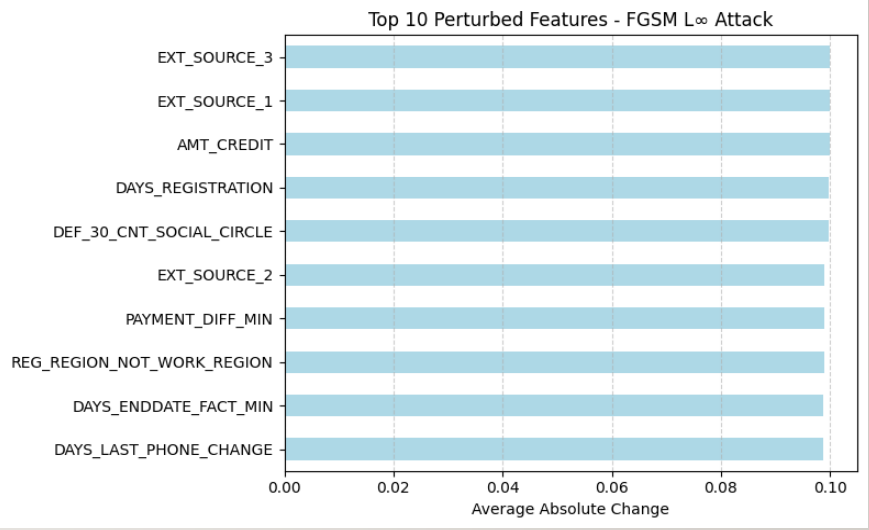
•**Class 0**: Precision = 0.79, Recall = 0.33, F1 = 0.46

•**Class 1**: Precision = 0.00, Recall = 0.01, F1 = 0.00

•**AUC = 0.0766**, **Accuracy = 0.3037**, **Overall F1 = 0.0034**

•**Top perturbed features**: EXT_SOURCE_3, EXT_SOURCE_2, STATUS_CANCELED_SUM_x, STATUS_1_SUM_SUM, AMT_DRAWINGS_OTHER_CURRENT_MEAN, NAME_CONTRACT_TYPE, REG_REGION_NOT_WORK_REGION, AMT_ANNUITY_MAX.

Insight: FGSM L∞ creates uniform, bounded changes that nearly destroy Class 0 recall (from 1.00 to 0.33). The model's reliance on external score features (EXT_SOURCE_2/3) and status indicators leaves it exposed to small but distributed input noise.



PGD L∞ Attack

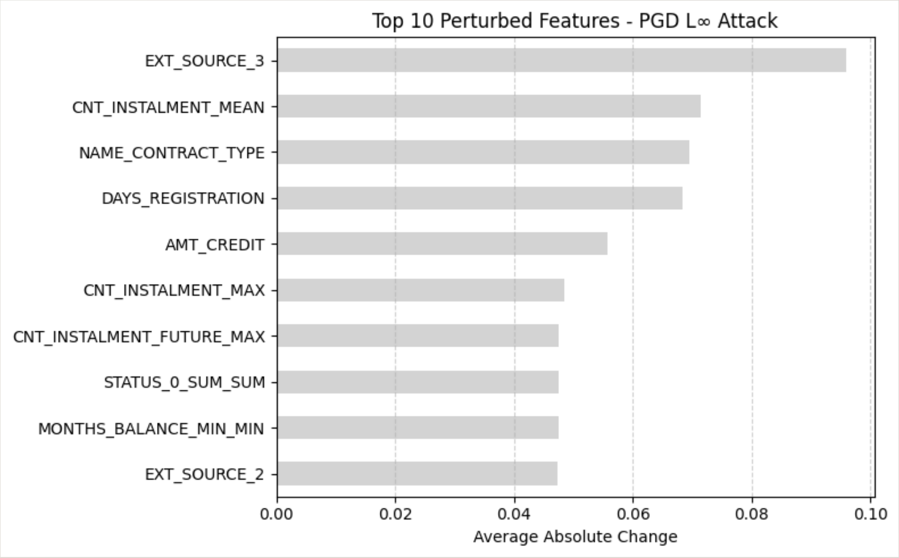
•**Class 0**: Precision = 0.79, Recall = 0.32, F1 = 0.46

•**Class 1**: Precision = 0.00, Recall = 0.00, F1 = 0.00

•**AUC = 0.0033**, **Accuracy = 0.2987**, **Overall F1 = 0.0003**

•**Top perturbed features**: EXT_SOURCE_3, CODE_GENDER, EXT_SOURCE_2, STATUS_CANCELED_SUM_x, NAME_EDUCATION_TYPE, FLAG_EMAIL, WALLSMATERIAL_MODE, STATUS_1_SUM_SUM.

•**Insight**: PGD L∞ pushes the model into complete failure. Class 0 barely functions, Class 1 is entirely lost, and AUC drops to near zero. The attack reveals that even stable-looking features like CODE_GENDER and EXT_SOURCE_3 are core to the model's fragile decision boundary.



OVERALL

The ROC curve highlights stark differences in model robustness under adversarial and covariate perturbations. While the clean baseline shows moderate separability (**AUC = 0.7729**), nearly all perturbations — especially adversarial — cause **severe degradation**.

•**L_0 attacks are highly disruptive**, with Gradient-Based L_0 dropping AUC to **0.0668**, the lowest among all.

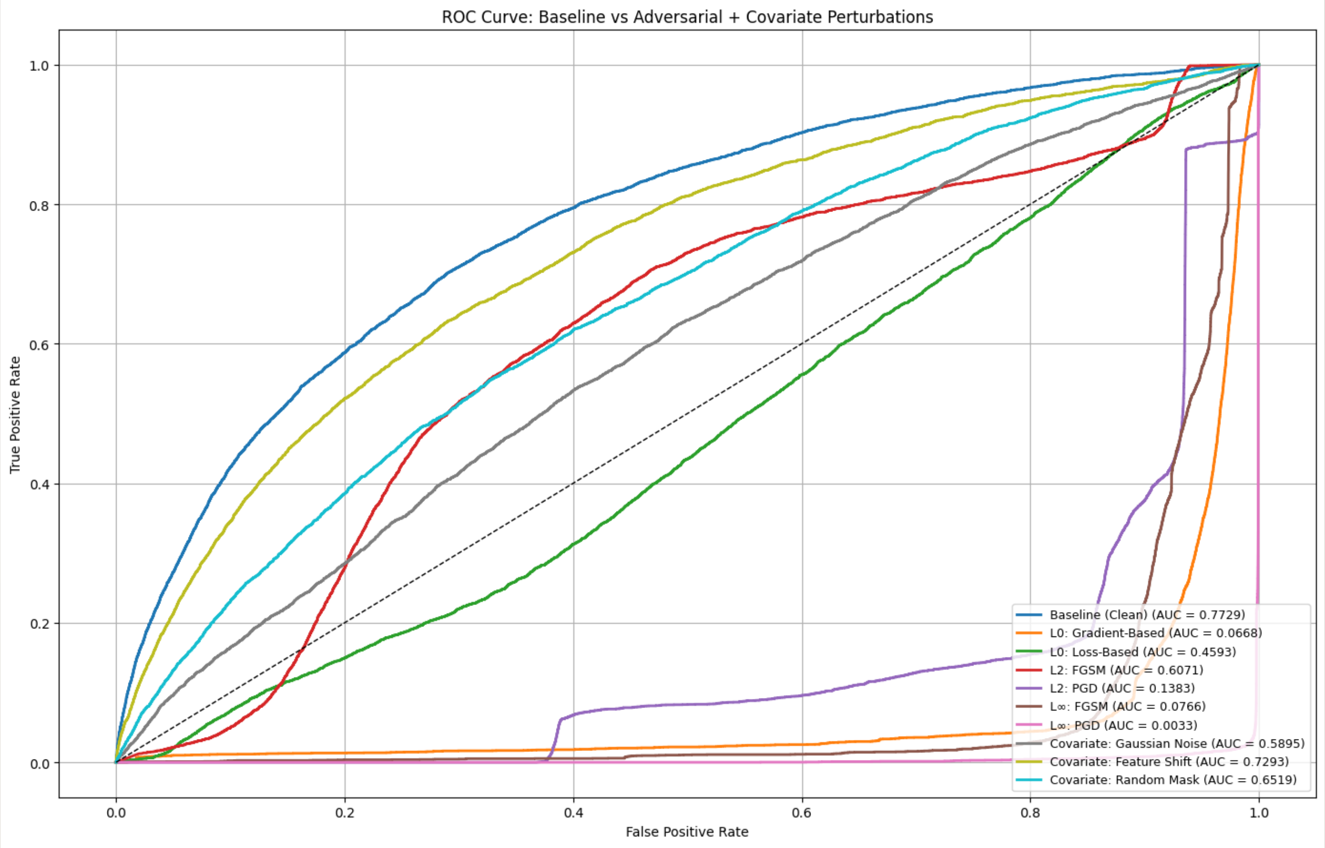
•**L_2 and L_∞ attacks also cause critical failure**:

- FGSM (L_2): **AUC 0.6071**
- PGD (L_2): **AUC 0.1383**
- FGSM (L_∞): **AUC 0.0766**
- PGD (L_∞): **AUC 0.0033**

•**Covariate perturbations are less harmful** but still impact performance:

- Feature Shift retains the most robustness (**AUC 0.7293**)
- Random Mask: **AUC 0.6519**
- Gaussian Noise: **AUC 0.5895**

Insight: The model is **extremely fragile to small, targeted feature changes**, especially in white-box adversarial settings. In contrast, it shows **some resilience to natural input drift**. This underscores the need for **robust training techniques**—including adversarial training or regularization—to guard against model collapse.



PROPOSED SOLUTIONS

Impact of SMOTE on TabNet Model Robustness

Objective:
To enhance model robustness by addressing class imbalance using SMOTE (Synthetic Minority Over-sampling Technique).

Key Results:

Insight:

SMOTE **dramatically improves Class 1 recall** (from near-zero to 0.57), helping the model finally recognize minority samples. However, this comes with a precision tradeoff for Class 1 and a small drop in Class 0 accuracy. Overall, the model is **more balanced and useful for detection**, especially in skew-sensitive tasks — making it a strong baseline before applying adversarial robustness techniques.

Metric	Baseline	SMOTE (1:1)
AUC	0.7729	0.7294
Accuracy	0.9200	0.7448
Class 1 F1 Score	0.1100	0.2636
Class 1 Recall	0.0600	0.5700
Class 1 Precision	0.4900	0.1700
Class 0 Recall	0.9900	0.7600
Class 0 Precision	0.9200	0.9500

PROPOSED SOLUTIONS

Impact of SMOTE on Tabnet Model Robustness

Metric	Baseline – Gaussian	SMOTE – Gaussian	Baseline – Feature Shift	SMOTE – Feature Shift	Baseline – Random Mask	SMOTE – Random Mask
AUC	0.5895	0.5107	0.7293	0.7095	0.6519	0.6285
Accuracy	0.8900	0.4400	0.9200	0.6700	0.9000	0.7500
Class 1 Recall	0.0600	0.6100	0.0100	0.6400	0.0700	0.3900
Prediction Change (%)	3.67	50.80	0.24	13.54	2.74	22.76

SMOTE significantly improves Class 1 recall under perturbations, making the model more responsive to minority signals. While this comes with increased prediction volatility—especially under Gaussian noise—it's a **worthwhile trade-off** compared to a blindly skewed baseline that ignores minority cases.

A more reactive model is **preferable**, as it engages with critical signals. However, this reactivity must be **refined**: by analyzing which features are most perturbed, we can **target regularization or training constraints** to control the model's response and **restore balance**. This allows us to retain sensitivity without sacrificing robustness.

PROPOSED SOLUTIONS

Impact of SMOTE on Tabnet Model Robustness

Attack Type	AUC (Baseline)	AUC (SMOTE)	F1 Score (Baseline)	F1 Score (SMOTE)	Class 1 Recall (Baseline)	Class 1 Recall (SMOTE)
L0 – Gradient	0.0668	0.4742	0.0117	0.0962	0.01	0.47
L0 – Loss	0.4593	0.3006	0.0390	0.0756	0.03	0.28
L2 – FGSM	0.6071	0.5987	0.1853	0.1431	0.37	0.72
L2 – PGD	0.1383	0.4406	0.0296	0.1131	0.15	0.55
L∞ – FGSM	0.0766	0.4526	0.0034	0.0929	0.01	0.49
L∞ – PGD	0.0033	0.0114	0.0003	0.0035	0.00	0.02

The SMOTE model demonstrates **greater robustness in capturing the minority class**, consistently achieving higher recall under all adversarial attacks. For example, in the Gradient L_0 attack, Class 1 recall rises from **0.01 (baseline)** to **0.47**, showing that SMOTE helps the model stay responsive even when inputs are perturbed. This robustness stems from the model's **increased exposure to minority patterns during training**, allowing it to generalize better to distorted samples.

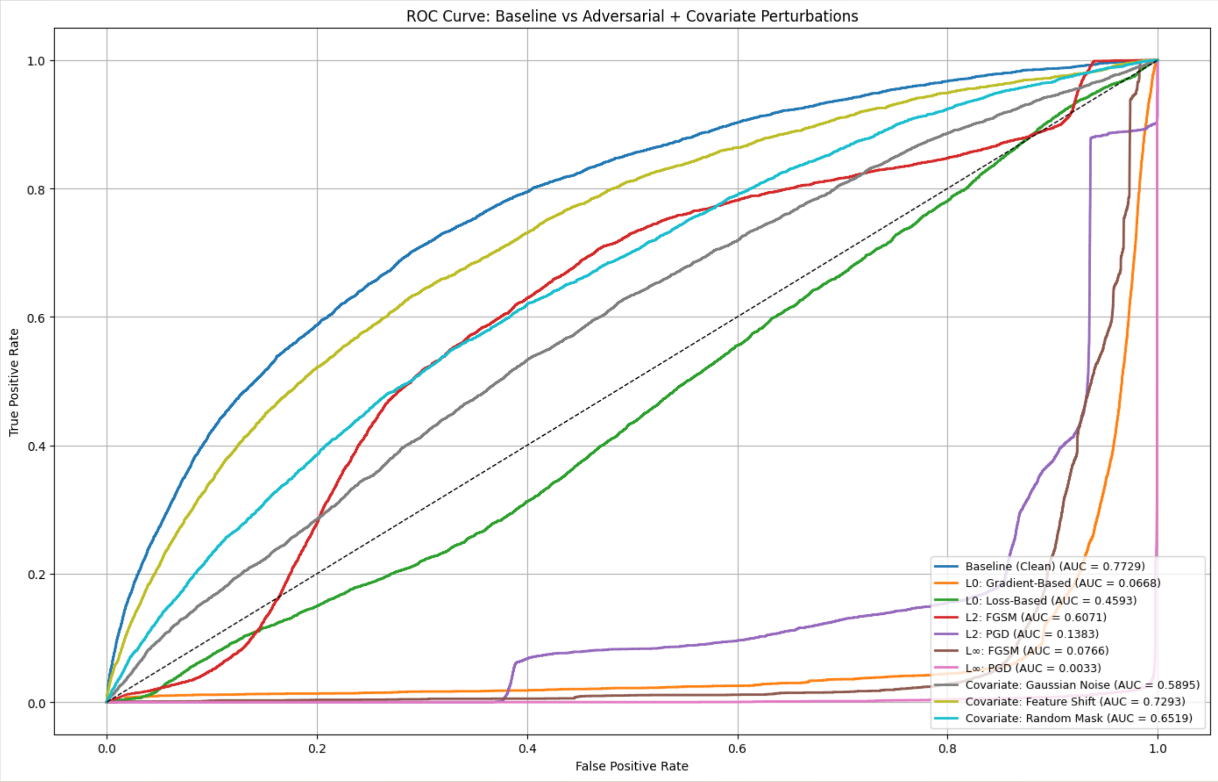
However, this improved sensitivity also makes the model **more reactive and less stable**, leading to lower precision and susceptibility to adversarial manipulation. To make this robustness truly reliable, it needs to be **tempered with control mechanisms**—such as adversarial training, input smoothing, or feature-level regularization—to **reduce overreaction while preserving minority detection**.

McGill

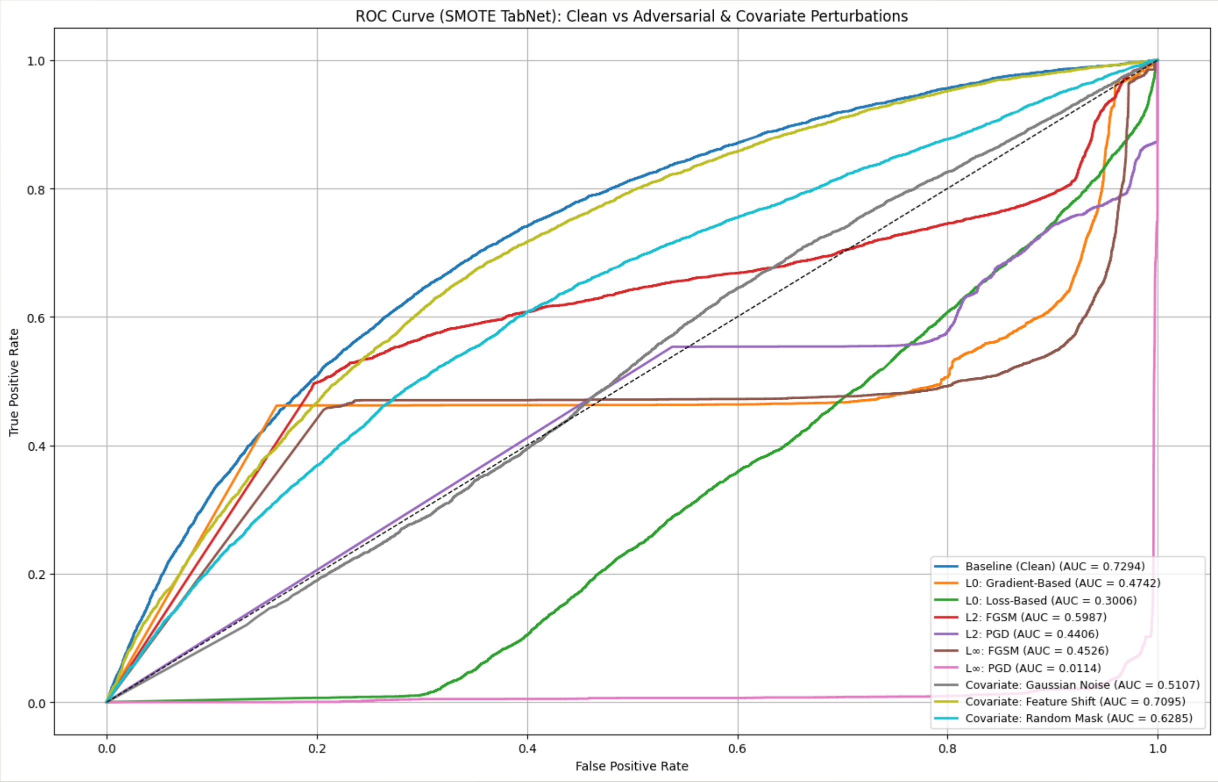
PROPOSED SOLUTIONS

Impact of SMOTE on Model Robustness

Before



After



PROPOSED SOLUTIONS

XGBoost vs. XGBoost + SMOTE

Baseline (Clean Data):

- XGBoost and XGBoost+SMOTE achieved similar AUC-ROC (~0.76), with SMOTE showing slightly lower recall on minority class (Class 1).

Under Perturbations:

Attack Type	AUC-ROC (XGB)	AUC-ROC (SMOTE)	Class 1 Recall (SMOTE)
Gaussian Noise	0.48	**0.58**	1.00 (but 99% prediction change)
Feature Shift	0.67	**0.70**	0.98 (low impact)
L0 Adversarial	0.18	↓ **0.12**	↓ 0.01

Key Insights:

- SMOTE slightly improves robustness under **feature shift**.
- It fails under **Gaussian noise** and **adversarial attacks**, showing degraded performance and extreme prediction instability.
- Class imbalance mitigation via SMOTE should be combined with **additional robustness techniques** to handle real-world perturbations effectively.

PROPOSED SOLUTIONS

TabNet with Noise Injection + Random Oversampling

Objective:

Enhance model robustness to class imbalance and input perturbations using Gaussian noise and RandomOverSampler.

Clean Performance:

- **AUC:** 0.7604 **Accuracy:** 72.4% **F1 Score:** 0.28
- Class 0: Precision 0.96 | Recall 0.73 | F1 0.83
- Class 1: Precision 0.18 | Recall 0.66 | F1 0.28

Adversarial Attack Resistance (AUC):

- Loss-Based L_0: **0.6643** (strongest)
- Gradient-Based L_0: 0.3071
- FGSM/PGD (L_2 & L_∞): ≤ 0.18

Robustness Under Covariate Perturbations:

- **Gaussian Noise:** AUC ↓ to 0.6989 | 21% prediction changes
- **Feature Shift:** AUC ↓ to 0.6593 | 34% prediction changes

Top Features:

EXT_SOURCE_2, EXT_SOURCE_3, AMT_BALANCE_MAX
→ Consistent influence across training and perturbation settings

While clean performance is strong and covariate robustness is moderate, the model remains vulnerable to adversarial attacks. Further defense strategies may be required.

PROPOSED SOLUTIONS

XGBoost + Gaussian Noise Injection

Clean Test Performance

Model	AUC	Class 1 F1 Score	Accuracy
Original XGBoost	0.735	0.11	91.8%
XGBoost + SMOTE	0.764	0.07	92.0%
XGBoost + Noise Injection	0.769	0.11	91.9%

Robustness to Covariate Perturbations

Model	Gaussian Noise AUC	Feature Shift AUC	% Prediction Changed
Original	0.57	0.64	~6.7%
SMOTE	0.58	0.70	~1.5%
Gaussian Noise	0.74	0.70	~1–2%

Adversarial L0 Attack

Model	AUC	Class 1 F1
Original	0.343	0.00
SMOTE	0.122	0.004
Gaussian Noise	0.646	0.057

Key Insight:

Gaussian Noise Injection improves overall robustness without sacrificing clean performance outperforming both the baseline and SMOTE model in all perturbation and attack scenarios.

EVALUATION TOOL FRAWORK

- **Included in submission**: One Python file with reusable robustness evaluation functions, One Documentation.
- **Designed for reuse**: Can be imported as a library into any model pipeline
- **Covers**: Perturbation-based testing (e.g., L0, L2, Gaussian noise, feature shift)
- **Plug-and-play**: Simple interface to evaluate and compare model stability
- **Planned but out of scope**: UX/UI component for visualizing evaluation results (pending time and full team availability)

QUESTIONS?

THANK YOU !

LUCAS DOAN & YVES ASSALI