Proactive Detection of Model Degradation in Financial Fraud Prediction with Delayed Labels

Akshay Sethi✉, Priyanshi Gupta, Sparsh Kansotia, Kamal Kant, and Nitish Srivasatava

AI Garage, Mastercard, India
{akshay.sethi, priyanshi.gupta, sparsh.kansotia, kamal.kant, nitish.srivasatava}@mastercard.com

Abstract. Financial fraud detection systems rely on machine learning models, but their performance degrades over time due to concept and covariate drift. A critical challenge is the delayed label problem: ground truth labels (confirming fraud) often arrive 1-6 months after the initial prediction. This creates a "blind period" where models can silently deteriorate, leading to substantial financial losses. Existing monitoring approaches, relying on delayed labels or statistical drift detection, are often too slow or insensitive. To address this, we propose PRODEM (PROactive DEtection of Model degradation), a framework that detects model degradation without immediate ground truth. PRODEM uses a meta-modeling technique: a sophisticated "meta-model" learns to predict when the deployed "primary" fraud model will make errors. We use a reverse distillation approach, where the meta-model specifically targets error prediction in out-of-time scenarios typical of fraud detection. Experiments on two proprietary datasets from a payment network show that PRODEM significantly improves degradation detection compared to statistical methods and recent drift detection techniques. Importantly, PRODEM identifies failing models before ground truth labels become available, mitigating the financial impact of model degradation in high-stakes decision-making. We also demonstrate PRODEM's effectiveness at identifying increases in false positive rates, a crucial but often overlooked aspect of fraud model monitoring.

Keywords: Fraud Detection · Delayed Labels · Model Degradation · Drift Detection · Meta-modeling · Reverse Distillation

1 Introduction

Financial fraud presents a significant threat to the global economy, with total losses soaring to approximately $485.6 billion in 2023 [7]. Fraud detection systems are essential for minimizing these financial losses and protecting institutions from evolving threats. However, a major challenge in this domain is the **delayed label problem**: ground truth labels confirming fraud often arrive 1-6 months after a transaction is processed [9]. This delay is due to factors such as lengthy investigations, customer dispute processes, and chargeback periods. This

"blind period" creates a significant vulnerability where machine learning models, commonly used for fraud prediction, can degrade silently due to covariate drift (changes in the input feature distribution while the relationship between features and target remains stable) and concept drift (changes in the relationship between features and the target variable).

These distribution shifts are driven by the constantly evolving landscape of financial fraud. Fraudsters adapt their tactics, new attack vectors emerge, and seasonal variations, economic fluctuations, changes in customer behavior all contribute to shifts in the underlying data distribution [38]. This makes the problem *increasingly critical* as fraud attacks become more sophisticated and regulatory scrutiny intensifies. Without timely feedback, a model trained on historical data can quickly become outdated and ineffective. The combination of sophisticated fraud techniques and stringent regulatory requirements necessitates effective monitoring systems.

Existing model monitoring approaches are inadequate for addressing this challenge effectively. Methods relying on delayed ground truth are, by definition, too late; significant losses can accumulate before any action is triggered. Statistical drift detection techniques, such as Kolmogorov-Smirnov tests [2], Population Stability Index (PSI) [6], and Wasserstein distance metrics [40] often struggle to distinguish between harmless covariate drift and performance-impacting concept drift. This can lead to either excessive false alarms, reducing operational efficiency by prompting unnecessary investigations, or, worse, missed degradation signals. More recent approaches integrate machine learning with statistical testing [20], but still exhibit similar limitations.

To address these shortcomings, we introduce PRODEM (PROactive DEtection of Model degradation), a framework that detects model degradation *before* ground truth labels become available. PRODEM employs a meta-modeling approach, where a sophisticated "meta-model" is trained to predict the errors of the deployed "primary" fraud prediction model. We leverage a **reverse distillation** technique, where, unlike traditional knowledge distillation—where a smaller model learns from a larger one—our meta-model is *more complex* than the primary model. This design choice, driven by production constraints on the primary model, enables the meta-model to capture subtle patterns indicative of future errors, particularly in out-of-time scenarios common in fraud detection with delayed feedback. The meta-model learns, in essence, the "failure modes" of the primary model. This proactive approach enables timely intervention – such as model retraining, feature re-engineering, or adjustments to decision thresholds – mitigating the financial impact of model degradation and preventing a prolonged period of increased losses.

A particularly overlooked aspect of monitoring financial fraud detection is the importance of identifying false positives, cases where legitimate transactions are incorrectly flagged as fraudulent. In label-delayed environments, a silent increase in false positive rates can go undetected for months, resulting in significant customer friction when legitimate transactions are declined and eventual loss of confidence in the automated system. PRODEM addresses this challenge by

outperforming existing methods in detecting increases in false positive rates with greater accuracy and timeliness.

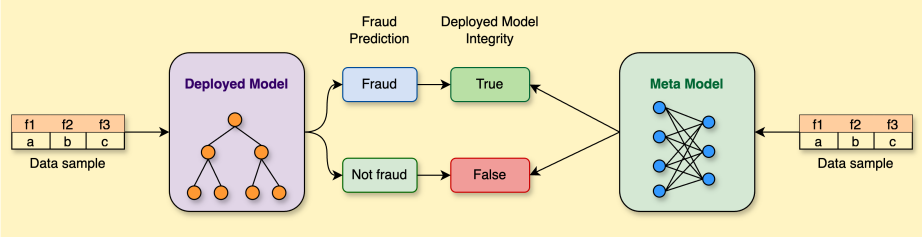

Fig. 1: The PRODEM framework architecture. The deployed model generates fraud predictions on transaction data, while the meta-model analyzes the same inputs to predict the likelihood of the deployed model making errors. This enables early detection of model degradation during periods of label unavailability.

Our key contributions are:

- A framework, PRODEM, for the proactive detection of model degradation in financial fraud detection systems operating under delayed label conditions, enabling timely intervention and risk mitigation.
- A meta-modeling approach leveraging reverse distillation to predict the errors of a deployed fraud model, specifically designed to handle the out-of-time challenges inherent in this domain. This allows the meta-model to learn complex patterns that the primary model misses.
- Demonstrated superior performance on two real-world financial fraud datasets from a payment network, significantly outperforming existing statistical and drift detection methods in identifying errors and model degradation.

The remainder of this paper is organized as follows: Section 2 reviews related work in model monitoring and drift detection. Section 3 describes our PRODEM framework in detail. Section 4 presents our experimental setup. Section 5 evaluates PRODEM's degradation detection capabilities and analyzes error prediction performance. Finally, Section 6 concludes the paper and outlines directions for future research.

2 Related Work

Model degradation from temporal distribution shifts presents a fundamental challenge in machine learning, particularly in dynamic domains like financial fraud detection. This section reviews relevant literature, highlighting limitations of existing approaches when confronted with delayed label availability.

2.1 Drift Detection

Drift detection methods identify changes in data distributions that may impact model performance [8]. These methods typically address two primary types of drift: covariate drift and concept drift. Traditional statistical approaches include the Kolmogorov-Smirnov test[2], Population Stability Index (PSI)[6,40], and Wasserstein distance metric [6,40], which compare reference and current distributions. However, these methods often fail to distinguish harmful concept drift from benign covariate drift, leading to false alarms or missed degradation signals.

Recent advances have integrated machine learning with statistical testing. Hinder et al. [16] combined JS-Divergence with feature importance analysis, while Webb et al. [37] introduced techniques for early drift detection by analyzing changes in low-density regions. Despite these improvements, these methods remain fundamentally reactive rather than proactive when confronted with delayed labels.

2.2 Model Monitoring Systems

Production model monitoring systems have evolved beyond simple performance tracking. Pozzolo et al. [30] developed adaptive learning frameworks for credit card fraud detection, though computational costs limit practical adoption [32]. Modern approaches employ multi-faceted monitoring stacks. Breck et al. [29] introduced a comprehensive validation framework examining data quality, model staleness, and prediction drift simultaneously.

These sophisticated systems remain largely reactive—they either observe performance degradation (requiring delayed labels) or detect statistical data drift (an imperfect proxy for performance degradation). They do not predict future model errors before ground truth labels become available.

2.3 Anomaly and Out of Distribution Detection

Anomaly detection and Out-of-Distribution (OOD) detection approaches identify irregular patterns and data points that deviate from expected distributions. Statistical anomaly detection methods like Isolation Forest [23] and One-Class SVM [36] establish decision boundaries around normal data. Deep learning approaches have expanded these capabilities, with methods such as Variational Autoencoders demonstrating particular effectiveness in practical applications like credit card fraud detection [33].

For OOD detection, Hendrycks and Gimpel [14] proposed Maximum Softmax Probability (MSP) for detecting OOD samples. Lee et al. [21] improved this approach with Mahalanobis distance-based confidence scores, while Liu et al. [24] demonstrated even better performance using energy-based models that leverage energy scores derived from logit outputs.

While these methods can signal model degradation through emerging anomaly clusters or distribution shifts, they have limitations for proactive monitoring.

They primarily focus on identifying individual anomalous or out of distribution instances rather than systematic degradation patterns, and don't directly assess whether model predictions will be incorrect.

2.4 Meta-Modeling for Error Prediction and Anomaly Detection

Our approach in PRODEM employs meta-modeling for error prediction, where auxiliary models predict the errors of primary fraud detection models. Several foundational works have explored this paradigm: Platanios et al. [28] developed a framework for estimating machine learning model accuracy without ground truth labels, while Raghu et al. [31] used meta-models to predict when deep learning models would fail in medical imaging classification tasks. Xiao et al. [39] proposed a meta-modeling framework for model error prediction in computer vision tasks, though their approach relies on dense feature representations specific to image data that do not transfer well to financial transaction data. Han et al. [12] propose SuperMentor, an oracle framework for predicting model correctness across in-domain, out-of-domain, and adversarial inputs with cross-model generalization capabilities, but their approach operates on static image datasets with readily available ground truth labels, making it unsuitable for delayed-feedback scenarios typical of fraud detection.

Another relevant paradigm comes from anomaly detection methods using reverse distillation [25,27,5,18]. In this approach, a "student" model is trained to reproduce the feature representations of a "teacher" model exposed only to normal data. When encountering anomalous samples, the student's inability to accurately reconstruct the teacher's output generates high reconstruction error, serving as an anomaly score. This principle of identifying deviations from learned norms aligns conceptually with our goal of predicting model errors when encountering evolving fraud patterns.

Building on these foundational ideas, PRODEM addresses the specific challenge of delayed labels in financial fraud detection through a novel meta-modeling approach with reverse distillation. Unlike traditional reactive methods that wait for confirmed fraud labels, our framework proactively identifies potential model failures before significant financial losses accumulate, enabling timely intervention in the face of evolving fraud patterns.

3 Methodology

3.1 Problem Formulation

Let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ represent the training dataset used to develop a fraud detection model, where each sample consists of tabular features $x_i \in \mathbb{R}^d$ and a corresponding binary fraud label $y_i \in \{0, 1\}$. The deployed fraud detection model $f_{\text{deploy}} : \mathbb{R}^d \to [0, 1]$ maps input features to fraud probability estimations, with a decision threshold θ determining the binary prediction: $\hat{y}_i = \mathbf{1}[f_{\text{deploy}}(x_i) > \theta]$.

In operational environments, financial institutions continuously receive new data $\mathcal{D}_{\text{recent}}$ requiring immediate predictions. However, the corresponding ground

truth labels only become available after a significant delay period τ (typically 1-6 months), creating a blind spot during which model degradation may occur undetected. This delay can be formalized as: y_t becomes available only at time $t + \tau$ where t represents the timestamp of an observation.

The objective of PRODEM is to develop a meta-model $f_{\text{meta}} : \mathbb{R}^d \to [0, 1]$ that estimates the probability that the deployed model will make an error on a given input x:

$$f_{\text{meta}}(x) = P(\hat{y} \neq y|x) = \begin{cases} 1, & \text{if } f_{\text{deploy}}(x) \text{ incorrectly predicts } y \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

By accurately estimating the deployed model's error probability distribution on recent data, the meta-model enables early detection of model degradation during the label lag period, allowing for timely interventions before significant financial losses materialize.

3.2 PRODEM Framework Overview

Our PRODEM framework, illustrated in Figure 1, addresses the challenge of detecting model degradation during the label lag period through a two-tier architecture:

Deployed Model The deployed model f_{deploy} represents the production fraud detection system in operation. In financial fraud detection domains, these models are predominantly tree-based ensemble algorithms (e.g., Gradient Boosted Trees [19,4], Random Forests[3]) due to their effectiveness with tabular data, interpretability requirements, and operational efficiency constraints. The deployed model remains fixed throughout the monitoring process, as it represents the actual production system under surveillance.

Meta-Model The meta-model f_{meta} serves as a specialized error predictor designed to anticipate the deployed model's misclassifications. Unlike the deployed model, the meta-model is not subject to the same operational constraints, enabling the utilization of more sophisticated architectures to capture nuanced patterns of model degradation. We implement the meta-model as a neural network with residual connections and attention mechanisms. This design allows f_{meta} to flexibly model complex error patterns that emerge from shifting data distributions, including evolving fraud strategies and gradual feature drift.

The meta-model processes the same input features as the deployed model but produces two outputs: (1) an approximation of the deployed model's prediction and (2) a probability estimate of the deployed model making an error.

This design enables PRODEM to provide actionable early warnings about reliability decay in the absence of fresh ground-truth labels, helping institutions mitigate risk before degradation materially impacts business performance. This dual output architecture is essential for the reverse distillation process described below.

3.3 Reverse Distillation Approach

Traditional knowledge distillation [11,17] transfers knowledge from a complex teacher model to a simpler student model. PRODEM inverts this paradigm through "reverse distillation", where a more sophisticated meta-model learns to model the behavior of a simpler deployed model with the specific objective of predicting its errors.

This approach is motivated by the operational reality in financial fraud detection, where production models must prioritize inference speed and interpretability over complexity. By allowing the meta-model to develop an internal representation of the deployed model's decision boundaries, we enable it to identify regions of uncertainty, blind spots, and failure modes that emerge as data distributions evolve over time.

3.4 Model Architecture

Deployed Model Architecture The deployed model typically utilizes tree-based ensemble methods such as XGBoost [4] or LightGBM [19], which are industry standards for fraud detection tasks. The specific architecture of the deployed model is not altered by PRODEM—instead, PRODEM treats it as a black box system that produces fraud probability scores.

Meta-Model Architecture The meta-model employs a neural network architecture with the following key components:

$$h_1 = \text{ReLU}(W_1 x + b_1) \tag{2}$$

$$h_i = h_{i-1} + \text{ReLU}(W_i h_{i-1} + b_i) \quad \text{for } i \in \{2, \ldots, L-1\} \tag{3}$$

$$z_{\text{meta}} = W_L h_{L-1} + b_L \tag{4}$$

$$p_{\text{meta}} = \sigma(W_{\text{err}} h_{L-1} + b_{\text{err}}) \tag{5}$$

where h_i represents the hidden layer activations, z_{meta} represents the logits mimicking the deployed model, and p_{meta} represents the probability of the deployed model making an error. The residual connections ($h_i = h_{i-1} + \ldots$) [13] facilitate gradient flow through deep networks, allowing the meta-model to learn complex representations that capture both the deployed model's behavior and its failure modes.

Additionally, we incorporate a self-attention mechanism [34] to enable the model to focus on feature interactions that are particularly relevant for error prediction:

$$Q = W_Q h_j, \quad K = W_K h_j, \quad V = W_V h_j \tag{6}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{7}$$

$$h_j^{\text{attn}} = \text{Attention}(Q, K, V) + h_j \tag{8}$$

This enables the meta-model to dynamically weight feature importance based on context, aiding detection of subtle patterns that precede model failures.

3.5 Training Methodology

Composite Loss Function The meta-model is trained using a composite loss function that balances two objectives:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{logit}} + (1 - \alpha)\mathcal{L}_{\text{error}} \tag{9}$$

where $\alpha \in [0, 1]$ is a hyperparameter controlling the trade-off between the two components.

The logit matching loss $\mathcal{L}_{\text{logit}}$ facilitates the meta-model's understanding of the deployed model's decision-making process:

$$\mathcal{L}_{\text{logit}} = \text{MSE}(z_{\text{deploy}}, z_{\text{meta}}) + \beta \cdot \text{KL}\left(\sigma\left(\frac{z_{\text{deploy}}}{T}\right) \middle\| \sigma\left(\frac{z_{\text{meta}}}{T}\right)\right) \tag{10}$$

where $\text{MSE}(\cdot, \cdot)$ is mean squared error, $\text{KL}(\cdot \| \cdot)$ is Kullback-Leibler divergence, $\sigma(\cdot)$ is the softmax function, z_{deploy} and z_{meta} are the logits from the deployed and meta-models respectively, T is a temperature parameter controlling distribution softness, and β is a balancing hyperparameter.

The error prediction loss $\mathcal{L}_{\text{error}}$ employs focal loss [22], an enhanced version of binary cross-entropy that addresses class imbalance by down-weighting easy-to-classify examples:

$$\mathcal{L}_{\text{error}} = -\gamma \cdot [c_{\text{true}} \cdot (1 - p_{\text{meta}})^{\gamma} \cdot \log(p_{\text{meta}}) + (1 - c_{\text{true}}) \cdot p_{\text{meta}}^{\gamma} \cdot \log(1 - p_{\text{meta}})] \tag{11}$$

where c_{true} is the ground truth correctness indicator, p_{meta} is the meta-model's predicted probability of error, and $\gamma \geq 0$ is the focusing parameter that reduces the loss contribution from easy examples.

Temporal Training Protocol To effectively address model degradation in out-of-time scenarios, we implement a rigorous temporal training protocol with structured chronological data partitioning. The complete dataset $\mathcal{D}$ is partitioned into three segments:

$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{meta}} \cup \mathcal{D}_{\text{test}} \quad \text{where} \quad t_{\text{train}} < t_{\text{meta}} < t_{\text{test}} \tag{12}$$

where t represents the timestamps associated with each partition.

- $\mathcal{D}_{\text{train}}$ (first N months): Used exclusively to train the deployed model
- $\mathcal{D}_{\text{meta}}$ (subsequent M months, typically $M < N$): Used to train the meta-model
- $\mathcal{D}_{\text{test}}$ (remaining available data): Used for evaluation

This temporal separation ensures that the deployed model f_{deploy} is trained on past data, reflecting a realistic production scenario, while the meta-model f_{meta} learns to identify errors arising from temporal distribution shifts. This setup allows for an accurate evaluation of the meta-model's ability to anticipate model degradation before it affects business metrics.

3.6 Degradation Detection Mechanism

Operational Metrics Estimation To operationalize model degradation detection, we construct a monitoring system based on meta-model error predictions. The meta-model enables estimation of critical performance metrics without requiring ground truth labels, providing early signals of degradation during the label lag period. We first estimate the confusion matrix components using meta-model predictions with a threshold ϵ:

$$\widehat{\text{FP}} = \sum_{i=1}^{|\mathcal{D}_{\text{recent}}|} \hat{y}_i \cdot \mathbb{I}[p_{\text{meta}}(x_i) > \epsilon] \tag{13}$$

$$\widehat{\text{FN}} = \sum_{i=1}^{|\mathcal{D}_{\text{recent}}|} (1 - \hat{y}_i) \cdot \mathbb{I}[p_{\text{meta}}(x_i) > \epsilon] \tag{14}$$

$$\widehat{\text{TP}} = \sum_{i=1}^{|\mathcal{D}_{\text{recent}}|} \hat{y}_i \cdot \mathbb{I}[p_{\text{meta}}(x_i) \leq \epsilon] \tag{15}$$

$$\widehat{\text{TN}} = \sum_{i=1}^{|\mathcal{D}_{\text{recent}}|} (1 - \hat{y}_i) \cdot \mathbb{I}[p_{\text{meta}}(x_i) \leq \epsilon] \tag{16}$$

Here, $\hat{y}_i$ is the deployed model's binary prediction, $p_{\text{meta}}(x_i)$ is the meta-model's predicted probability of error for input x_i, and $\mathbb{I}[\cdot]$ is the indicator function that returns 1 if the condition is true and 0 otherwise. Using these estimated confusion matrix components, we calculate key performance metrics that financial institutions prioritize:

$$\widehat{\text{VDR}} = \frac{\sum_{i:\hat{y}_i=1} \text{Amount}_i \cdot \mathbb{I}[p_{\text{meta}}(x_i) \leq \epsilon]}{\sum_i \text{Amount}_i \cdot y_i} \tag{17}$$

$$\widehat{\text{FAR}} = \frac{\widehat{\text{FP}}}{\widehat{\text{TP}}} \tag{18}$$

where VDR (Value Detection Rate) measures the proportion of total fraud value correctly identified by the model, and FAR (False Alarm Rate) quantifies the ratio of false positives to true positives. In fraud detection, due to significant class imbalance, we monitor estimated False Positives ($\widehat{\text{FP}}$) directly as a degradation signal. Since true negatives (TN) vastly outnumber other components and remain relatively stable, an increase in false positives serves as an effective proxy for model degradation without requiring the full FPR calculation. The threshold ϵ is typically calibrated using historical data to optimize the trade-off between detection sensitivity and specificity.

Degradation Detection Heuristics We establish degradation detection heuristics based on these estimated metrics by comparing them against baseline performance values:

$$\text{Degradation Alert} = \begin{cases} 1, & \frac{\widehat{\text{VDR}}_t}{\widehat{\text{VDR}}_{\text{baseline}}} < \delta_{\text{VDR}} \quad \text{or} \\ & \frac{\widehat{\text{FAR}}_t}{\widehat{\text{FAR}}_{\text{baseline}}} > \delta_{\text{FAR}} \quad \text{or} \\ & \frac{\widehat{\text{FP}}_t}{\widehat{\text{FP}}_{\text{baseline}}} > \delta_{\text{FP}} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where δ_{VDR}, δ_{FAR}, and δ_{FP} are configurable thresholds (typically set to 0.75, 1.25, and 1.25 respectively) that trigger alerts when VDR decreases significantly or when FAR or FP increases significantly compared to the baseline.

4 Experiments

In this section, we present a thorough evaluation of the PRODEM framework on two financial fraud detection datasets, showcasing its ability to identify model degradation early — before ground truth labels become available.

4.1 Experimental Setup

Datasets We evaluate PRODEM on two proprietary datasets from a payment network company.

- **Transaction Fraud Dataset (TF-Dataset)**: This dataset contains card transaction data from a large issuer spanning 9 months with over 3 million transactions per month. Approximately 50% of transactions are declined by the issuer bank through their risk management systems. The feature set includes diverse risk scores, merchant identifiers, transaction type indicators, velocity features, and historical transaction patterns. The fraud rate among approved transactions is 0.42%. The dataset exhibits significant temporal patterns in both feature distributions and fraud tactics.
- **Account Fraud Dataset (AF-Dataset)**: This dataset focuses on account-level (card-level) fraud detection for a large issuer spanning 9 months. It contains approximately 5 million active accounts per month with a positive class rate of 0.91%. Features include account profile characteristics, aggregated transaction patterns, risk scores, spending behavior statistics, and historical dispute information. This dataset exhibits pronounced seasonal effects and regional fraud pattern evolution.

Temporal Data Partitioning Following the temporal protocol described in Section 3, we partition each dataset as follows. For both the TF and AF-Datasets, we use months 1-2 for $\mathcal{D}_{\text{train}}$ (deploy model training), month 3 for $\mathcal{D}_{\text{meta}}$ (meta-model training), and months 4-9 for $\mathcal{D}_{\text{test}}$ (evaluation).

The months 4-9 for $\mathcal{D}_{\text{test}}$ have been referred to as months 1-6 in the results section for better readability and ease of explanation.

4.2 Baselines

We compare PRODEM against established approaches for model monitoring and drift detection:

Statistical Distribution Monitoring

- **Kolmogorov-Smirnov (KS) Test**: Applied to top 20 features by importance to detect univariate distribution shifts.
- **Kullback-Leibler (KL) Divergence**: Measures distribution shifts in continuous features.

Output Distribution Monitoring

- **Maximum Class Probability (MCP)** [15]: Monitors the maximum prediction probability of the deployed model. Predictions with MCP below a threshold are flagged as potential errors.
- **Class Probability Entropy (CPE)** [26]: Calculates the entropy of the prediction probability distribution. Higher entropy indicates greater uncertainty in the model's prediction, potentially signaling an error.

Implementation Details

- **Fraud Detection Model ($f_{\textbf{deploy}}$)**: XGBoost trained on $\mathcal{D}_{\text{train}}$ with 150 trees (max depth=10, learning rate=0.05) and subsampling (row=0.8, column=0.8). Hyperparameters optimized via optuna [1] using 5-fold cross-validation and weighted log-loss to address class imbalance (1:99 for TF-Dataset, 1:55 for AF-Dataset).
- **Meta-Model**: FT-Transformer [10] with tokenized tabular inputs and a Transformer encoder with multi-head self-attention. Architecture: 4 transformer blocks (hidden sizes: 256, 128, 64, 32), residual connections, layer normalization, SeLU activations, dropout (0.2), and L2 regularization (0.001). Optimized using Adam (lr=0.001) with cosine annealing, composite loss from Section 3 ($\alpha = 0.3$, $T = 2.0$), batch size 4096, trained for 100 epochs with early stopping, implemented in PyTorch and trained on NVIDIA A100 GPUs (80GB).
- **Baselines**: Baselines include Kolmogorov-Smirnov (KS) Test ($p < 0.01$), Kullback-Leibler (KL) Divergence ($KL > 0.5$), Maximum Class Probability (MCP) ($MCP < 0.7$), and Class Probability Entropy (CPE) ($CPE > 0.8$).

5 Results and Analysis

5.1 Error Prediction Performance

We first evaluate its ability to accurately predict when the deployed model will make errors. This error prediction capability forms the foundation of our proactive degradation detection framework. Table 1 presents the meta-model's performance in predicting deployed model errors across the two datasets, compared against baseline approaches.

Table 1: Error prediction performance comparison across datasets (best results in **bold**)

Dataset	Method	EDP	EDR
TF-Dataset	MCP	0.43	0.37
	CPE	0.51	0.48
	PRODEM	**0.66**	**0.64**
AF-Dataset	MCP	0.42	0.41
	CPE	0.45	0.49
	PRODEM	**0.68**	**0.65**

It's important to note that traditional statistical techniques (KS Test and KL Divergence) are not included in this comparison as they detect feature or score distribution drift rather than predicting specific model errors at the instance level. Their detection mechanisms operate at a distributional level, which is fundamentally different from the error prediction task evaluated here.

The results demonstrate that PRODEM achieves significant improvements in both precision (EDP) and recall (EDR) of error detection compared to uncertainty-based methods (MCP and CPE). For the TF-Dataset, PRODEM achieves an error detection precision of 0.66 and recall of 0.64, representing improvements of 29.4% and 33.3% respectively over the best baseline method. Similar improvements are observed for the AF-Dataset, with PRODEM achieving 51.1% higher precision and 32.7% higher recall compared to the best baseline.

This superior error prediction capability stems from the meta-model's ability to learn specific error patterns of the deployed model through our reverse distillation approach, enabling more accurate identification of potential misclassifications in the complex financial fraud domains represented by our proprietary datasets.

5.2 Model Degradation Detection

We next analyze how PRODEM's error prediction capability translates into early detection of model degradation compared to baseline approaches. Table 2 summarizes the performance of PRODEM and baseline approaches, showing the

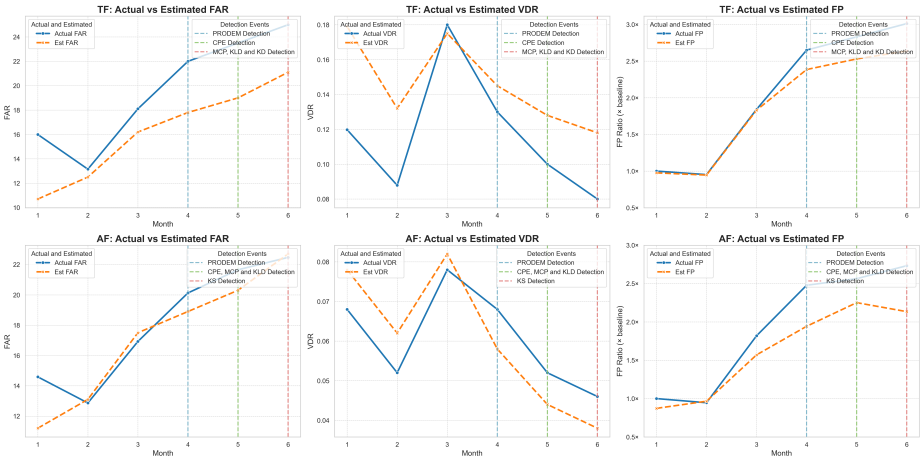

Fig. 2: Temporal comparison of estimated (PRODEM) versus actual performance metrics (VDR, FAR, and false positives) for TF-Dataset and AF-Dataset over the six-month testing period. Vertical markers indicate the month of degradation detection for different methods, with PRODEM demonstrating earlier detection (month 4) compared to baseline approaches.[1]

Table 2: Model degradation detection timing across datasets (lower month number indicates earlier detection)

Method	**TF-Dataset (Month)**	**AF-Dataset (Month)**
KS Test	6	6
KL Divergence	6	5
MCP	6	5
CPE	5	5
PRODEM	**4**	**4**

month in which each method first identified significant performance deterioration.

The results demonstrate that PRODEM provides significantly earlier detection of model degradation compared to baseline methods. For the TF-Dataset, PRODEM detected degradation in Month 4, while the best baseline methods only identified issues in Month 5. For the AF-Dataset, PRODEM detected degradation in Month 4, a full month before any baseline approach.

5.3 Temporal Performance Monitoring

Finally, we analyze how PRODEM monitors model performance over time compared to the actual performance metrics derived from ground truth labels. This analysis is crucial for understanding how accurately PRODEM can track degradation patterns in deployed models throughout the label delay period. Figure 2

shows the estimated versus actual performance metrics for both datasets over the testing period.

For the TF-Dataset, our analysis reveals significant changes in key metrics over the six-month monitoring period. The value detection rate (VDR) showed substantial fluctuations, with a 55.6% decrease from Month 1 to Month 6. PRODEM effectively tracked these changes, with estimated VDR following a similar pattern, showing a 34.1% decline over the same period. Most critically, false positives increased dramatically by 201% from Month 1 to Month 6. PRODEM successfully estimated this trend, projecting a 170% increase in false positives over the same timeframe. The false acceptance rate (FAR) estimates by PRODEM showed a consistent upward trend, increasing by 56% from Month 1 to Month 6.

For the AF-Dataset, we also observed notable variations in key metrics. The VDR decreased by 32.4% from Month 1 to Month 6. PRODEM accurately tracked this degradation, with estimated VDR decreasing by 51.3%. False positives increased dramatically by 173% over the monitoring period. PRODEM's estimates closely followed this trend, projecting a 144% increase in false positives. The FAR estimates showed a steady increase of 48% from Month 1 to Month 6.

For both datasets, PRODEM demonstrated remarkable accuracy in tracking critical metric changes during periods of significant degradation. The average deviation between PRODEM's estimated metrics and actual metrics was approximately 14.5% for VDR and 7.2% for false positives throughout the testing period. This confirms PRODEM's effectiveness in providing advance warning of model degradation, with particularly strong performance in tracking false positive rate increases, a critical concern in fraud detection systems where the cost of false positives directly impacts customer experience and operational efficiency.

6 Conclusion

We introduced PRODEM, a proactive framework for detecting model degradation in financial fraud detection systems under label delay constraints. Leveraging a meta-modeling approach with reverse distillation, PRODEM identifies performance deterioration without requiring immediate ground truth. Evaluations on two proprietary datasets show PRODEM provides a 1–2 month advance warning over traditional monitoring, enabling timely intervention. Key contributions include a meta-modeling approach for proactive error prediction of fraud models, a reverse distillation-based loss, and a temporal training protocol for out-of-time degradation. PRODEM significantly improves error detection precision and recall over baselines.

Future work includes incorporating explainability, and developing adaptive monitoring thresholds based on the severity and type of detected degradation patterns.

[1] To preserve data confidentiality, the Y-axis values for False Positives are presented as ratios normalized to the baseline month (Month 1).

7 Ethics Statement

Our proposed framework does not raise any ethical concerns. However, it is essential to acknowledge that ethical applications of financial fraud detection can greatly benefit from the improved early warning capabilities and performance enhancements provided by PRODEM. To ensure responsible and socially beneficial deployment of machine learning in financial systems, it is crucial to exercise caution, transparency, and fairness in model monitoring and adaptation. This involves not only maintaining accountability in how alerts are generated and acted upon, but also ensuring that model updates do not reinforce existing biases or disproportionately impact specific customer segments over time or across different demographics.

References

1. Akiba, T., Sano, S., et al.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2623–2631 (2019)
2. Berger, V.W., Zhou, Y.: Kolmogorov–smirnov test: Overview. Wiley statsref: Statistics reference online (2014)
3. Breiman, L.: Random forests. Machine learning (2001)
4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 785–794 (2016)
5. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9737–9746 (2022)
6. Dries, A., Rückert, U.: Adaptive concept drift detection. Statistical Analysis and Data Mining: The ASA Data Science Journal (2009)
7. Friedman, A.: Global financial report, 2024. NASDAQ (2024)
8. Gama, J.a., Žliobaitundefined, I., et al.: A survey on concept drift adaptation. ACM Comput. Surv. (2014)
9. Garg, N.: Feature engineering for fraud detection. Fennel (2022)
10. Gorishniy, Y., Rubachev, I., et al.: Revisiting deep learning models for tabular data. Advances in neural information processing systems (2021)
11. Gou, J., Yu, B., et al.: Knowledge distillation: A survey. International Journal of Computer Vision (2021)
12. Han, S., et al.: An oracle for in-domain, out-of-domain, and adversarial errors. arXiv preprint (2024)
13. He, K., Zhang, X., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
14. Hendrycks, D., Basart, S., et al.: Scaling out-of-distribution detection for real-world settings. arXiv preprint (2019)
15. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint (2016)
16. Hinder, F., Vaquet, V., Hammer, B.: Feature-based analyses of concept drift. Neurocomputing (2024)
17. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint (2015)

18. Jiang, Y., Cao, Y., Shen, W.: A masked reverse knowledge distillation method incorporating global and local information for image anomaly detection. Knowledge-Based Systems (2023)
19. Ke, G., Meng, Q., et al.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems (2017)
20. Kulatilleke, G.K.: Challenges and complexities in machine learning based credit card fraud detection. arXiv preprint (2022)
21. Lee, K., Lee, K., et al.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Advances in neural information processing systems (2018)
22. Lin, T.Y., Goyal, P., et al.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
23. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 8th IEEE international conference on data mining. pp. 413–422 (2008)
24. Liu, W., Wang, X., et al.: Energy-based out-of-distribution detection. Advances in neural information processing systems (2020)
25. Liu, X., Wang, J., et al.: Unlocking the potential of reverse distillation for anomaly detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 5640–5648 (2025)
26. Macêdo, D., Ren, T.I., et al.: Entropic out-of-distribution detection. IEEE Transactions on Neural Networks and Learning Systems (2021)
27. Nasser, S.A., Gupte, N., Sethi, A.: Reverse knowledge distillation: Training a large model using a small one for retinal image matching on limited data. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 7778–7787 (2024)
28. Platanios, E., et al.: Estimating accuracy from unlabeled data: A probabilistic logic approach. Advances in neural information processing systems (2017)
29. Polyzotis, N., Zinkevich, M., et al.: Data validation for machine learning. Proceedings of machine learning and systems (2019)
30. Pozzolo, A.D., Bontempi, G.: Adaptive machine learning for credit card fraud detection (2015)
31. Raghu, M., Blumer, K., et al.: The algorithmic automation problem: Prediction, triage, and human effort. arXiv preprint (2019)
32. Sethi, T.S., Kantardzic, M., Arabmakki, E.: Monitoring classification blindspots to detect drifts from unlabeled data. In: 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI). pp. 142–151 (2016)
33. Tingfei, H., Guangquan, C., Kuihua, H.: Using variational auto encoding in credit card fraud detection. IEEE Access (2020)
34. Vaswani, A., Shazeer, N., et al.: Attention is all you need. Advances in neural information processing systems (2017)
35. Veit, A., Wilber, M.J., et al.: Residual networks behave like ensembles of relatively shallow networks. Advances in neural information processing systems (2016)
36. Vert, R., Vert, J.P., Schölkopf, B.: Consistency and convergence rates of one-class svms and related algorithms. Journal of Machine Learning Research (2006)
37. Webb, G.I., Hyde, R., et al.: Characterizing concept drift. Data Mining and Knowledge Discovery (2016)
38. Wiki: Wirecard scandal. Fennel (2020)
39. Xiao, T., Xia, T., et al.: Learning from massive noisy labeled data for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
40. Yurdakul, B.: Statistical properties of population stability index. Western Michigan University (2018)