UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

PARAMETER-FREE LOSS FOR IMBALANCED CLASSIFICATION IN MEDICARE FRAUD DETECTION

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements

for the Degree of

Master of Science

BY

DAVID COLINDRES

Norman, Oklahoma

2026

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/342535

PARAMETER-FREE LOSS FOR IMBALANCED
CLASSIFICATION IN MEDICARE FRAUD DETECTION

A THESIS APPROVED FOR THE
SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Talayeh Razzaghi, Chair

Dr. Shivakumar Raman

Dr. Byeong-Min Roh

Dr. Kash Barker

© Copyright by DAVID COLINDRES 2026
All Rights Reserved.

Acknowledgements

This thesis would not exist in its current form without the people acknowledged.

I owe the direction of my self-pioneered research to my advisor, Dr. Talayeh Razzaghi, who initially suggested utilizing PF-loss in Medicare fraud detection as a thesis topic and trusted me to find my own path through it. She gave me the freedom to explore methods entirely new to me while offering precise, detail-oriented guidance that kept my work focused; I left each conversation with clarity and confidence.

My thanks go out to Dr. Shivakumar Raman, Dr. Kash Barker, and Dr. Byeong-Min Roh for serving on my committee and for asking the questions that strengthened both my reasoning, defense, and final publication. The final version of this thesis is significantly better because of their collective input.

My gratitude goes out to the faculty in the Industrial & Systems Engineering program for making my accelerated master's possible. Their teaching in various ISE-core topics provided me with the technical foundation on which this thesis was built. The accelerated B.S./M.S. track demanded a pace that often felt overwhelming, but it taught me how to define a large project, build thoroughly, and carry it through to completion.

To my family, thank you for your patience, belief, and prayers during this process, even during the periods when it felt impossible to start and make progress on such a complex topic. Your support will always mean the world to me. To the friends and colleagues who offered encouragement, perspective, or even distraction at the right moments, thank you. That balance of both adventure and academia mattered more than you know. The world is your playground.

Sincerely,

David Colindres

David Colindres

Contents

Acknowledgements iv

Preface viii

1 Introduction 1

1.1 Traditional Fraud Detection Limitations 1

1.2 Machine Learning for Fraud Detection 2

1.3 The Class Imbalance Problem 3

1.4 Research Limitations 3

1.5 Thesis Contributions 4

1.6 Thesis Organization 5

2 Literature Review 6

2.1 Overview of Fraud Detection Research 6

2.2 Machine Learning Models Used in Fraud Detection 7

2.3 Data Reduction Techniques for Imbalanced Big Data 9

2.4 Process-Based and Sequence Mining Approaches 9

2.5 Imbalance Learning Techniques 9

2.6 Cost-Sensitive Loss Functions 10

2.7 Dataset Comparison Across Studies 12

2.8 Literature Gaps 12

3 Methodology 14

3.1 Overall Framework 14
3.2 Neural Network Architecture 15
3.3 Loss Functions 16
3.4 Model Architectures 19
3.5 Hyperparameter Tuning 20
3.6 Evaluation Metrics 25
3.7 Experimental Design 26
3.8 Quantitative Risk Analysis 26
4 Data Description 28
4.1 Data Sources 28
4.2 Data Merging and Feature Engineering 29
4.3 Missing Data Analysis 30
4.4 Data Partitioning 31
4.5 Descriptive Statistics 31
4.6 Comparative Model Breakdown 33
4.7 Quantitative Risk Analysis 33
5 Computational Results 35
5.1 Cross-Validation Framework 36
5.2 Model Performance 36
5.3 PF-Loss vs. Baseline Losses 40
5.4 Class Imbalance Strategy Comparison 40
5.5 Training Time Analysis 42
5.6 SHAP Interpretability 43
5.7 Quantitative Risk Analysis 45
5.8 Median-Based Expenditure Threshold Heuristic 48

6 Discussion 50
6.1 Interpretation of Results 50
6.2 SHAP Interpretability Insights 51
6.3 Limitations 52
6.4 Implications and Contributions 52
7 Conclusion 53
7.1 Summary of Findings 53
7.2 Key Contributions and Practical Implications 53
7.3 Future Work 54
7.4 Final Remarks 54
Bibliography 55

Appendices **60**

Detailed Variable Description Table 61
PF-Loss Implementation Pseudocode 65

Preface

Most of my time as an Industrial & Systems Engineering student has been spent challenging myself with various complex topics. I entered the accelerated program with a strong interest in quantitative finance and in understanding how mathematical models can reveal structure in noisy, fast-paced environments. When my advisor proposed healthcare fraud detection as a thesis topic, the domain was unfamiliar, but the underlying problem was not. It was a problem of rare events hidden within a large, imbalanced dataset. The cost of misclassification is uneven, and the value of a model depends on what it labels correctly.

That framing is what made the project compelling to me. Medicare fraud costs the U.S. healthcare system tens of billions of dollars each year, yet fraudulent providers represent only a small share of all claims. This creates exactly the kind of class imbalance that conventional classifiers often handle poorly. This thesis asks whether the Parameter-Free Loss function, a recently proposed method for training deep learning models on imbalanced data, can narrow the gap between neural networks and well-tuned traditional classifiers on tabular healthcare claims. Answering that question required evaluating twelve models under multiple imbalance-handling strategies and grounding the results in both statistical testing and a quantitative risk framework. The scope of that process pushed me well beyond any single course or prior project.

This work taught me more than I expected. It taught me about CMS claims data, the operational realities of fraud investigation, and the discipline required to build a reproducible experimental pipeline at scale. It also taught me the importance of results, especially when they contradict expectations. I hope this thesis reflects both the rigor of that process and a genuine curiosity about how these methods will advance future research and innovation.

Abstract

Healthcare fraud costs the U.S. Medicare system over $100 billion annually; yet, fraudulent providers represent only a small share of all claims, creating a severe class imbalance that conventional classifiers overlook. This thesis presents one of the first applications of Parameter-Free Imbalance-Aware Loss (PF-Loss) for deep learning to Medicare fraud detection and evaluates it alongside eleven other supervised models on a publicly available CMS dataset containing 5,410 providers, 558,211 claims, and 163 engineered features. The study benchmarks eight traditional and ensemble machine learning classifiers and four tabular ResNet variants, each distinguished by its loss function: PF-Loss (PF), Weighted Cross-Entropy (WCE), Focal Loss (FL), and Weighted Label-Distribution-Aware Margin Loss (WLDAM). All models are evaluated under a unified experimental framework with G-mean as the primary performance metric.

PF-Loss achieved the highest G-mean among all models (86.66%), outperforming the strongest traditional classifier, XGBoost (85.26%), as well as the other neural network variants. All four ResNet loss variants exceeded the traditional models in G-mean, which indicates that deep learning combined with imbalance-aware loss design is effective for tabular healthcare fraud detection. PF-Loss completed training in 19.7 seconds, making it 3.7 to 11.5 times faster than the other neural network variants, because its parameter-free formulation required only three architecture configurations, compared with 12, 27, and 36 for WCE, Focal Loss, and WLDAM, respectively. A systematic class imbalance analysis further showed that traditional models deteriorated sharply under the original no-resampling condition, whereas PF-Loss maintained a G-mean of 86.66% without external resampling; combining PF-Loss with SMOTE improved performance only marginally to 86.74%.

SHAP-based interpretability analysis identified provider-level and physician-level reimbursement amounts, claim counts, admission duration, and deductible amounts as the most consistent fraud indicators across model families, indicating that fraud predictions were driven primarily by billing behavior rather than patient demographics. This thesis also extends classification outputs into a quantitative risk framework based on Basel II/III credit risk methodology, including probability calibration, expected loss estimation, budget-constrained audit optimization formulated as a 0–1 knapsack problem, and Monte Carlo simulation for tail-risk assessment. The results from this study show that PF-Loss is an effective and efficient alternative to conventional imbalance strategies, especially for tabular healthcare fraud detection. The model offers strong predictive performance, a significantly lower tuning burden, and practical application for resource-constrained fraud investigation units.

Keywords: Medicare fraud detection, Parameter-free loss, Class imbalance, Tabular ResNet, G-mean, SHAP interpretability, Cost-sensitive learning, Deep learning, Quantitative risk analysis

1 Introduction

Healthcare financial fraud imposes a substantial economic burden on the United States healthcare system. The National Health Care Anti-Fraud Association (NHCAA) estimates that fraud accounts for a minimum of 3% of total healthcare expenditures, while U.S. government and law enforcement agencies report it to be as high as 10% of annual healthcare spending [1]. With total U.S. healthcare expenditures exceeding $4.3 trillion, even more conservative estimates from the Centers for Medicare and Medicaid Services (CMS) imply losses above $100 billion annually [2]. These losses divert resources away from legitimate patient care, medical research, and healthcare infrastructure. In turn, the issues increase financial pressure on government/insurance agencies and healthcare institutions.

Within Medicare, the scale of financial exposure is especially important. Medicare serves as the primary health insurance provider for Americans aged 65 and older, selectively serving younger individuals with disabilities and covering over 65 million beneficiaries while processing billions of claims annually [3]. Although improper payments include both intentional fraud and unintentional billing errors, the scale of this estimate illustrates the financial vulnerability of a system that processes enormous volumes of claims through automated judgment and limited reviews of payments.

Such cases also highlight the difficulty of detecting fraudulent behavior through manual review. The forms of Medicare fraud vary and are becoming increasingly sophisticated. The NHCAA identifies common schemes, which include upcoding, unbundling, phantom billing, and kickback arrangements tied to fraudulent referrals [1]. Johnson and Khoshgoftaar [4] documented a notable COVID-19-era scheme: bundling office visit fraud, in which providers billed for in-person procedures, especially COVID-19 tests, that were never performed.

Beyond financial loss, healthcare fraud may also produce non-financial harm. Medically unnecessary procedures expose patients to avoidable risks, while medical identity theft can introduce incorrect diagnoses and treatments into patient records [1]. Federal enforcement actions recover billions of dollars annually, and the Department of Justice (DOJ) claims that some of these recoveries still represent only a small fraction of estimated losses [5]. Across Oklahoma, similar concerns appear, as reports indicate increasing worry over anomalous billing patterns and overcharges from providers [6]. Overall, the need for scalable and data-driven fraud detection systems becomes increasing.

1.1 Traditional Fraud Detection Limitations

Traditional fraud detection approaches have relied primarily on rule-based expert systems and manual audits. Although both remain useful in some settings, they face important limitations as healthcare systems increase in size and complexity.

Rule-based systems identify suspicious claims by flagging predefined billing thresholds,

coding irregularities, or policy violations. Static rules are fixed, unchanging guidelines that do not adjust to changing circumstances or new information. Therefore, while these systems are straightforward to implement, they are often difficult to adapt when new fraud strategies do not match existing patterns. Fraudulent providers may also adjust their behavior to remain below thresholds that flag fraud, reducing the effectiveness of preventative measures. Specifically, new fraud patterns must be identified, formalized, validated, and integrated into the system before they can be detected in practice. This timely process often leads to high false positive rates and an excessive auditing burden [7, 8].

Manual audits provide a second line of defense by allowing trained investigators to review claims, billing histories, and provider documentation. However, manual review is expensive and difficult to scale in a system that processes billions of claims annually. Detection may also occur months or years after the fraudulent behavior began, allowing losses to accumulate before corrective action is taken [9].

A further limitation relates to fraud being statistically anomalous. Fraud is inherently rare relative to legitimate activity, which means that a naive classifier can achieve high overall accuracy by predicting the majority class while failing to detect any fraud whatsoever. Thus, meaningful fraud detection requires machine learning methods that explicitly address class imbalance rather than optimizing for aggregate accuracy only [10].

1.2 Machine Learning for Fraud Detection

Machine Learning has emerged as a leading approach for fraud detection, as it provides flexible, data-driven alternatives to static rule systems. In supervised learning, models learn decision boundaries directly from labeled training examples and can capture complex relationships that are difficult to specify manually. In the Medicare setting, the task is to map provider billing profiles to a binary fraud label using historical claims data and administrative labels. This framework allows models to identify subtle patterns that distinguish fraudulent from legitimate billing behavior and to adapt as fraud patterns change over time [11, 12].

Machine Learning approaches to fraud detection span a range of model families, each offering different advantages for structured healthcare data. Linear models remain useful baseline methods, given their interpretability and efficiency, while tree-based ensemble methods often produce the strongest results on tabular fraud datasets. Other approaches, including support vector machines and deep learning architectures, have also shown value, especially when complex patterns or large-scale data are involved. Altogether, this body of work suggests that Medicare fraud detection benefits from models capturing nonlinear billing behavior, while also highlighting that performance depends not only on model choice but also on the methods used to treat class imbalance.

1.3 The Class Imbalance Problem

Class imbalance occurs when the class of interest is substantially underrepresented relative to the majority class. In Medicare fraud detection, this imbalance can be severe. Hancock et al. [13] reported fraud rates below 0.1% in large Part B and Part D datasets. Although the provider-level dataset used in this thesis has a higher fraud prevalence after aggregation, at around 9.35%, it remains strongly imbalanced from a classification perspective. Under these conditions, models trained with standard cross-entropy often favor majority-class performance and fail to identify important fraud cases [10, 14].

Two broad strategies are commonly used to address class imbalance. The first is resampling, which alters the training distribution through oversampling or undersampling [15]. SMOTE (Synthetic Minority Over-sampling Technique) generates synthetic minority-class observations by interpolating between existing minority examples and is one of the most widely used oversampling methods. Although effective in various settings, oversampling can increase the risk of overfitting, whereas undersampling may discard useful majority-class information [16]. Lopo and Hartomo [17] found that SMOTE improved healthcare insurance fraud detection performance under several evaluation metrics.

The second strategy is loss-function modification. Cost-sensitive methods such as Weighted Cross-Entropy (WCE) [18], Focal Loss (FL) [19], and Weighted Label-Distribution-Aware Margin Loss (WLDAM) [20] all attempt to increase attention to minority-class errors. However, each introduces one or more hyperparameters that must be tuned carefully, which can become costly on large datasets.

PF-Loss (PF) [21] takes a different approach. Instead of relying on manually specified class-imbalance hyperparameters, PF-Loss replaces each sample's predicted probability in the standard cross-entropy objective with the average predicted probability of its class within the current mini-batch. This shifts the loss from sample-level emphasis to class-level performance and automatically increases attention to classes with lower average predicted probabilities. PF-Loss provides class-rebalancing behavior without the need for additional hyperparameter tuning required by methods such as WCE, FL, or WLDAM, as well as various other ML models.

1.4 Research Limitations

Despite prior work on imbalanced classification in healthcare fraud detection and similar areas, several crucial gaps remain.

First, no prior study has applied PF-Loss within deep learning to Medicare fraud detection data or to tabular healthcare fraud classification. Although PF-Loss has shown strong performance on imbalanced image-classification benchmarks, its effectiveness on tabular healthcare data with mixed feature types, structured missingness, and severe imbalance has

not been established.

Second, Medicare fraud detection studies rarely provide comprehensive benchmarking across multiple machine learning model families and loss-function variants within a unified experimental framework. Third, SHAP-based interpretability remains underused in this literature despite its value for understanding model behavior and supporting auditing decisions. Fourth, the interaction between resampling strategies and cost-sensitive loss functions has not been systematically explored in this setting.

1.5 Thesis Contributions

This thesis makes the following contributions to machine learning for healthcare fraud detection:

1. **Application of PF-Loss to Medicare fraud detection.** This study is among the first to apply Parameter-Free Loss (PF-Loss) to Medicare fraud detection. PF-Loss is implemented within a designated neural network architecture, ResNet (which refers to the 4 deep learning loss-functions: PF, WCE, FL, and WLDAM), and evaluated alongside eight additional models on CMS provider-level fraud data, demonstrating its viability for tabular healthcare binary classification.

2. **Comprehensive benchmarking framework.** A total of twelve models are evaluated under a unified experimental framework, including eight traditional and ensemble machine learning methods and four deep learning loss-function variants. All models are trained and validated using consistent preprocessing, evaluation procedures, and G-mean as the primary performance metric.

3. **Reproducible preprocessing pipeline.** A comprehensive, reproducible data pipeline is developed to integrate various features: beneficiary demographics, inpatient and outpatient claims, and provider fraud labels, among many others. The pipeline includes feature engineering to generate 163 provider-level variables, one-hot encoding, Min-Max normalization, and SMOTE-based resampling within a leakage-free pipeline architecture.

4. **SHAP-based interpretability analysis.** SHapley Additive exPlanations (SHAP) is a game-theoretic method of interpreting machine learning model results by quantifying how much each feature contributes to specific predictions. SHAP values are used to identify the most consistent fraud-related features across model families, providing an interpretable view of model behavior and offering practical insights for CMS auditors.

5. **Training efficiency and quantitative risk analysis.** Beyond predictive performance, the thesis evaluates PF-Loss' computational efficiency in comparing training duration with alternative loss functions (which require additional hyperparameter tuning). The study also extends classification outputs into a quantitative risk framework that includes probability calibration, Brier score evaluation, dollar-denominated expected loss

estimation, budget-constrained audit optimization, and Monte Carlo simulation for risk assessment.

1.6 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 3 presents the literature review on fraud detection research, machine learning models, class imbalance learning, PF-Loss, and hyperparameter optimization. Chapter 4 describes the methodology, including the prediction pipeline, the mathematical formulation of PF-Loss, benchmark loss functions, and model architectures. Chapter 5 presents the data sources, the feature engineering process, the missing data analysis, and an exploratory data analysis. Chapter 6 reports the computational results, including model comparisons, loss-function analysis, and SHAP-based feature importance, among others. Chapter 7 discusses the findings in relation to prior work and practical deployment. Chapter 8 concludes the thesis and outlines directions for future research.

2 Literature Review

Fraud detection is prevalent across finance, insurance, healthcare, and governments alike. This chapter reviews the literature most relevant to the thesis topic addressed: fraud detection methods, machine learning models, data reduction techniques, class-imbalance learning, dataset comparisons, and the key gaps that motivate the application of Parameter-Free Loss (PF-Loss) to provider-level Medicare fraud detection.

2.1 Overview of Fraud Detection Research

The study of fraud detection itself has greatly evolved over the past three decades. Early work relied primarily on statistical anomaly detection and rule-based expert systems [22]. As fraud tactics became more sophisticated, machine learning emerged as a more adaptable alternative. Bolton and Hand [23] provided a foundational taxonomy distinguishing supervised approaches (such as labeled fraud examples) from unsupervised approaches; they help identify anomalous patterns without labels. Travaille et al. [24] framed Medicare fraud detection as a supervised binary classification task at the provider level, using the List of Excluded Individuals and Entities (LEIE) as the source of fraud labels.

Later published work expanded the application of features through more sophisticated data engineering. Herland et al. [8] showed that integrating multiple CMS data sources improved performance relative to individual sources, achieving an AUC of 0.816 using Logistic Regression. They also proposed an approach in which bottlenecks between a physician's predicted and recorded specialty, inferred from procedure profiles, serve as fraud signals. On a larger Medicare dataset containing 5.3 million samples, Leevy et al. [25] reported that CatBoost achieved the highest AUPRC of 0.8124 and argued that AUPRC is more appropriate than ROC-AUC under severe class imbalance.

Johnson and Khoshgoftaar [4] directed attention to the data preparation pipeline, estimating that 80% of the machine learning workflow is devoted to data understanding and preparation. Using CMS claims data from 2013 to 2019, they constructed multiple labeled datasets, introduced provider-level summary features, and proposed cumulative labeling, which improved fraud detection performance by 15%. This demonstrated how conventional cross-validation can yield overly optimistic results, with k-fold NPI-based cross-validation recommended instead.

More recently, Hancock et al. [26] proposed an unsupervised labeling and feature selection framework for Medicare data with a fraud rate of only 0.069%, combining SHAP-based feature ranking with autoencoder-generated labels to support supervised learning for class imbalance.

Systematic Reviews

Two recent systematic reviews provide comprehensive coverage of the field. Du Preez, Bhattacharya, Beling, and Bowen [27] analyzed 137 papers from 2000 to 2024 using a PRISMA-compliant methodology. Their review confirmed that provider fraud is the most frequently studied category, that CMS datasets are the primary benchmark (71 of 137 studies), and that Random Forest (RF), neural networks, and Gradient Boosted Trees (GBT) are among the most commonly used models. Importantly, their review does not discuss PF-Loss or other parameter-free loss functions, which adds novelty to this thesis.

Curtis, Billion-Polak, Khoshgoftaar, and Furht [28] surveyed 22 classification techniques published between 2017 and 2024, identifying five major gaps: limited statistical significance testing, weak inter-model benchmarking, underdeveloped explainability, little use of transfer or incremental learning, and limited reproducibility. The present thesis addresses three of these directly through comprehensive benchmarking, SHAP-based explainability, and reproducibility with other datasets. As with du Preez et al. [27], none of the methods reviewed by Curtis et al. apply PF-Loss.

Qualitative and Empirical Context

A complementary body of work has examined the structural factors that make Medicare vulnerable to fraud. Thorpe, Deslich, Sikula, and Coustasse [29] regard Medicare as a 'pay-and-chase' system in which claims are paid before meaningful review occurs. This notes that imprecision in billing codes allows rejected claims to be resubmitted under alternate codes. Coustasse, Layton, Nelson, and Walker [30] documented the large losses associated with upcoding in hospital and emergency department billing, emphasizing the importance of reimbursement-related features in fraud detection models. Pande and Maas [31] examined 318 physicians placed on the Office of Inspector General (OIG) Exclusion List and found that the average fraud amount per convicted physician was approximately $1.4 million, with many continuing to practice after conviction. Hill, Hunter, Johnson, and Coustasse [32] described early CMS data mining approaches based on geographic outliers and regression-based anomalies, which can be viewed as precursors to modern machine learning methods.

2.2 Machine Learning Models Used in Fraud Detection

Jordan and Mitchell [33] argue that supervised learning has become prominent in practical AI applications. This is because function approximation from labeled training data is both theoretically well understood and computationally low-maintenance. Medicare fraud detection fits naturally within this framework. Different model families capture different structures in the data, which motivates the comparison of various models in this thesis.

Linear Models

Regularized Logistic Regression remains a widely used baseline, given its interpretability, probabilistic output, and computational efficiency [23]. Within healthcare fraud detection, it performs competitively with strong feature engineering, although its linear structure limits its ability to model complex nonlinear relationships [7].

Tree-Based Ensemble Methods

Tree-based ensemble methods have become central to fraud detection on structured data. Random Forest (RF) [34] builds an ensemble of decision trees on random subsets and aggregates predictions through majority voting. Gradient Boosting (GB) methods build trees sequentially, with each new tree trained to reduce the errors made by its predecessors. XGBoost (XGB) [35] provides regularization and efficiency on sparse data; CatBoost (CB) [36] addresses target leakage through ordered boosting; LightGBM [37] improves speed through multi-directional growth; and AdaBoost (AB) [38] iteratively re-weights observations based on prior errors.

Hancock et al. [13] provide the most comprehensive comparison of gradient-boosted decision tree classifiers in the context of Medicare fraud, consistently finding that CatBoost yields the strongest performance across both Part B and Part D datasets. Hancock, Bauder, Wang, and Khoshgoftaar [39] further propose an ensemble supervised feature selection method that substantially reduces dimensionality while maintaining or improving AUPRC. These findings compel the inclusion of gradient boosting models in the benchmarking framework of this thesis.

Support Vector Machines

Support Vector Machines (SVM) classify observations by identifying the maximum-margin hyperplane between classes [40]. Both linear and nonlinear kernel variants have been applied to fraud detection [41, 42]. However, SVMs scale poorly to large datasets and are sensitive to class imbalance [43], making them less practical for large-scale Medicare applications.

Deep Learning

Deep neural networks learn hierarchical feature representations through multiple composed nonlinear modules [44]. In Medicare fraud detection, Johnson and Khoshgoftaar [14] found that relatively shallow architectures performed well on CMS Part B data when combined with suitable imbalance handling, a finding that informed the ResNet neural network architecture used in this thesis. Autoencoders provide a complementary approach by training to reconstruct normal behavior and flagging instances with high reconstruction error [45]. Mayaki and Riveill

[46] proposed MINN-AE, combining a multiple-input network with an LSTM autoencoder, and Hancock et al. [26] extended this idea by using reconstruction error to rank instances by anomaly severity.

Hyperparameter Optimization

Bergstra and Bengio [47] showed that random search is more efficient than grid search when only a subset of hyperparameters meaningfully affect performance. Bayesian optimization improves further through probabilistic surrogate models [48], as noted by Bennett et al. [42] applied the Tree Parzen Estimator in clinical prediction settings. HyperBand [49] combines exploration with early stopping, and BOHB (Bayesian Optimization and HyperBand) [50] integrates Bayesian optimization with HyperBand. In this thesis, CV grid search is used for traditional models with small search spaces, while structured search with early stopping is employed for deep learning.

2.3 Data Reduction Techniques for Imbalanced Big Data

Hancock et al. [13] provide the most systematic study to date of data reduction techniques for Medicare fraud detection, focusing on Random Undersampling (RUS) and ensemble supervised feature selection. Their analysis showed that a 1:81 undersampling ratio often performed comparably to the full data with substantial computational savings, and that models trained on 10 to 15 selected features often outperformed those trained on the full feature set. A major contribution is their systematic comparison of five experimental scenarios combining RUS and feature selection, evaluated through ANOVA and Tukey HSD post-hoc testing. CatBoost consistently achieved the strongest performance across scenarios.

2.4 Process-Based and Sequence Mining Approaches

A complementary line of work examines sequential billing patterns at the transaction level. Matloob et al. [51] proposed a sequence mining architecture using the PrefixSpan algorithm to flag deviations from expected service sequences within hospital specialties. This approach differs from provider-level classification but is complementary; future systems combining both strategies may better detect fraud at a granular level.

2.5 Imbalance Learning Techniques

Class imbalance is one of the central methodological challenges in fraud detection. Existing approaches fall into two categories: data-level resampling and algorithm-level cost-sensitive learning.

Resampling Methods

SMOTE. SMOTE (Synthetic Minority Over-sampling Technique) [15] is a widely used oversampling method for imbalanced classification. It creates synthetic minority-class observations by interpolating between a sample and its k nearest neighbors. This expands the minority-class region without exact duplication, but it can introduce noise when minority samples lie near class boundaries or overlap with the majority class [16]. Despite this limitation, SMOTE remains a standard baseline in healthcare fraud detection and has shown strong performance in prior studies [14, 17].

Random Undersampling. Random Undersampling (RUS) reduces class imbalance by removing majority-class observations until a target class ratio is reached. Its main advantage is computational efficiency, since it reduces both training time and memory requirements. However, it may discard useful majority-class information and weaken the representation of the non-fraud decision boundary [10]. Hancock et al. [13] found that RUS often achieved competitive performance on Medicare fraud data while providing substantial computational savings.

Random Oversampling. Random Oversampling (ROS) addresses class imbalance by duplicating existing minority-class observations until the desired class ratio is achieved. Unlike SMOTE, it does not generate synthetic data, which makes it simple to implement and broadly applicable across feature types. Its main limitation is the increased risk of overfitting, since models may memorize repeated minority-class samples rather than learn generalizable patterns [10]. Even so, ROS remains a useful baseline and has shown competitive performance in prior Medicare fraud studies [14].

2.6 Cost-Sensitive Loss Functions

Weighted Cross-Entropy Loss

Weighted cross-entropy (WCE) applies class-specific weights to each sample's cross-entropy contribution. Weights are set using the inverse class-frequency heuristic, with a weight-scaling factor α_{scale} serving as the primary tuning parameter [18].

Focal Loss

Focal loss (FL) [19] modifies cross-entropy by down-weighting well-classified examples and concentrating learning on harder cases near the decision boundary. Two hyperparameters must be tuned: the focusing parameter γ and the class-balancing parameter α.

Weighted LDAM Loss

Weighted LDAM (WLDAM) [20] imposes class-specific margins in logit space, with the motivation that minority classes require larger margins for better generalization. The margin for class c is defined as $\Delta_c = C/N_c^{0.25}$, where C is a tunable constant. In practice, both C and the weight-scaling factor α_{scale} require tuning.

Parameter-Free Loss

Parameter-Free Loss (PF) was introduced by Du et al. [21] to eliminate the need for loss-weight hyperparameter tuning. Instead of relying on each sample's predicted probability, PF-Loss uses the average predicted probability of the corresponding class via mini-batch:

$$\text{PF} = -\frac{1}{M}\sum_{j=1}^{M}\log(q_j), \qquad q_j = \frac{1}{N_j}\sum_{i=1}^{N_j} p_i \tag{1}$$

where M is the number of classes, N_j is the number of samples in class j within the current mini-batch, and p_i is the predicted probability of correct classification for sample i. This shifts the loss from sample-level emphasis to class-level performance, automatically equalizing class importance without manually tuned imbalance hyperparameters. Du et al. evaluated PF-Loss on image datasets CIFAR-10, CIFAR-100, and Fashion-MNIST under imbalance ratios of up to 200, reporting improvements in G-mean and AUC relative to WCE and Focal Loss. Its application to tabular healthcare data had not been investigated before this work.

Li et al. [52] proposed the related but distinct Parameter-Free Extreme Learning Machine (PF-ELM), which assigns smaller weights to higher-loss observations at the sample level. While both methods are parameter-free, PF-ELM operates at the sample level, whereas PF-Loss operates at the class level through mini-batch averages, making PF-Loss better suited for deep learning. Other loss functions for imbalanced learning, including DR-loss [53], AP-loss [54], Mixed Entropy loss for graph data [55], and automated loss-design methods such as AutoLINC [56] and AutoBalance [57], continue to expand methodologies but still require at least one tuning parameter.

SHAP-Based Interpretability

SHAP (SHapley Additive exPlanations), introduced by Lundberg and Lee [58], assigns an attribution value to each feature of each prediction. SHAP satisfies local accuracy, missingness, and consistency, and TreeSHAP makes exact explanations feasible for tree ensembles. Van den Broeck et al. [59] showed that SHAP is #P-hard in the general case, calling for the need for efficient implementations for applied fraud detection.

2.7 Dataset Comparison Across Studies

Standardized benchmark datasets for Medicare fraud detection remain limited. The Kaggle dataset used in this thesis includes 5,410 providers with a fraud prevalence of approximately 9.35%. At a larger scale, Herland et al. [8] integrated multiple CMS sources into a dataset of approximately 760,000 instances, with fraud rates ranging from 0.038% to 0.074%. Johnson and Khoshgoftaar [4] constructed enriched datasets from CMS claims spanning 2013 to 2019, achieving AUC values between 0.9485 and 0.9571 using XGBoost. Leevy et al. [25] used CMS prescriber data with 5.3 million records and a 0.0692% fraud rate, while Hancock et al. [13] used aggregated Part B and Part D datasets with 80 to 82 features. These differences in scale, feature construction, and labeling strategy make direct comparison across studies difficult and further motivate a unified benchmarking framework.

2.8 Literature Gaps

The preceding review identifies several important gaps that this thesis addresses.

1. PF-Loss has not yet been applied to Medicare fraud detection data or any tabular healthcare classification problems. Although PF-Loss has shown strong results in image-classification, its effectiveness on tabular healthcare data with mixed feature types, structural missingness, and severe class imbalance has not been established.

2. Comprehensive benchmarking remains limited. Most prior studies evaluate only a small number of models under different preprocessing pipelines and evaluation settings. None of the reviewed studies examine multiple model types and loss-function variants within a unified experimental framework.

3. SHAP-based interpretability analysis combined with systematic class imbalance strategy comparisons remains underutilized. While SHAP is increasingly common in applied machine learning, its consistent use across multiple fraud detection model families has received limited attention.

4. The combination of cost-sensitive loss functions with various resampling techniques, such as SMOTE, random undersampling, random oversampling, and both, has not been systematically explored in this context.

5. Prior work has not translated Medicare fraud classification outputs into a quantitative risk framework that supports expected loss estimation, budget-constrained audit optimization, and Monte Carlo simulation.

Table 1: Summary of representative literature in ML-based fraud detection

Study	**Model(s)**	**Metrics**	**Imbalance Technique**	**Application**
[7]	C4.5, LR, RF	AUC, TPR	RUS	Medicare claims
[8]	RF, LR, MLP	AUC	RUS	Medicare (multi-source)
[14]	DNN	AUC, G-mean	RUS, ROS, SMOTE, CS	Medicare claims
[4]	XGB, RF	AUC, TPR, G-mean	Cumul. labeling, NPI-CV	Medicare claims
[39]	CB, XGB, LightGBM	AUPRC	Ens. feat. sel.	CMS Part B/D
[13]	CB, XGB, RF, ET, LightGBM	AUPRC	RUS, feat. sel.	CMS Part B/D
[26]	AE, IF, DT, RF	AUPRC	Unsup. AE labeling	Medicare claims
[25]	CB, ET, XGB, RF, LR	AUC, AUPRC	Binary vs. OCC	Credit card, Medicare
[27]	Survey of 137 studies	Various	RUS, SMOTE, ROS	Healthcare (systematic review)
[28]	Survey of 22 techniques	Various	Various	Healthcare (survey)
[21]	ResNet, MLP	G-mean, AUC	PF-Loss	CIFAR, Fashion-MNIST

Abbreviations: AE = Autoencoder; AUC = Area Under ROC Curve; AUPRC = Area Under Precision-Recall Curve; CB = CatBoost; CS = Cost-Sensitive; Cumul. = Cumulative; DNN = Deep Neural Network; DT = Decision Tree; Ens. = Ensemble; ET = Extra Trees; feat. sel. = feature selection; G-mean = Geometric Mean; IF = Isolation Forest; LR = Logistic Regression; MLP = Multilayer Perceptron; NPI-CV = NPI-based Cross-Validation; OCC = One-Class Classification; RF = Random Forest; ROS = Random Over-Sampling; RUS = Random Under-Sampling; SMOTE = Synthetic Minority Over-sampling Technique; SVM = Support Vector Machine; TPR = True Positive Rate; Unsup. = Unsupervised; XGB = Extreme Gradient Boosting.

3 Methodology

This chapter presents the prediction framework used for Medicare provider fraud detection with Parameter-Free Loss (PF-Loss). The methodology integrates demographic variables, claim-level records, and behavioral billing patterns extracted from public CMS datasets. It contributes to a unified pipeline that supports systematic comparison across supervised fraud detection models.

3.1 Overall Framework

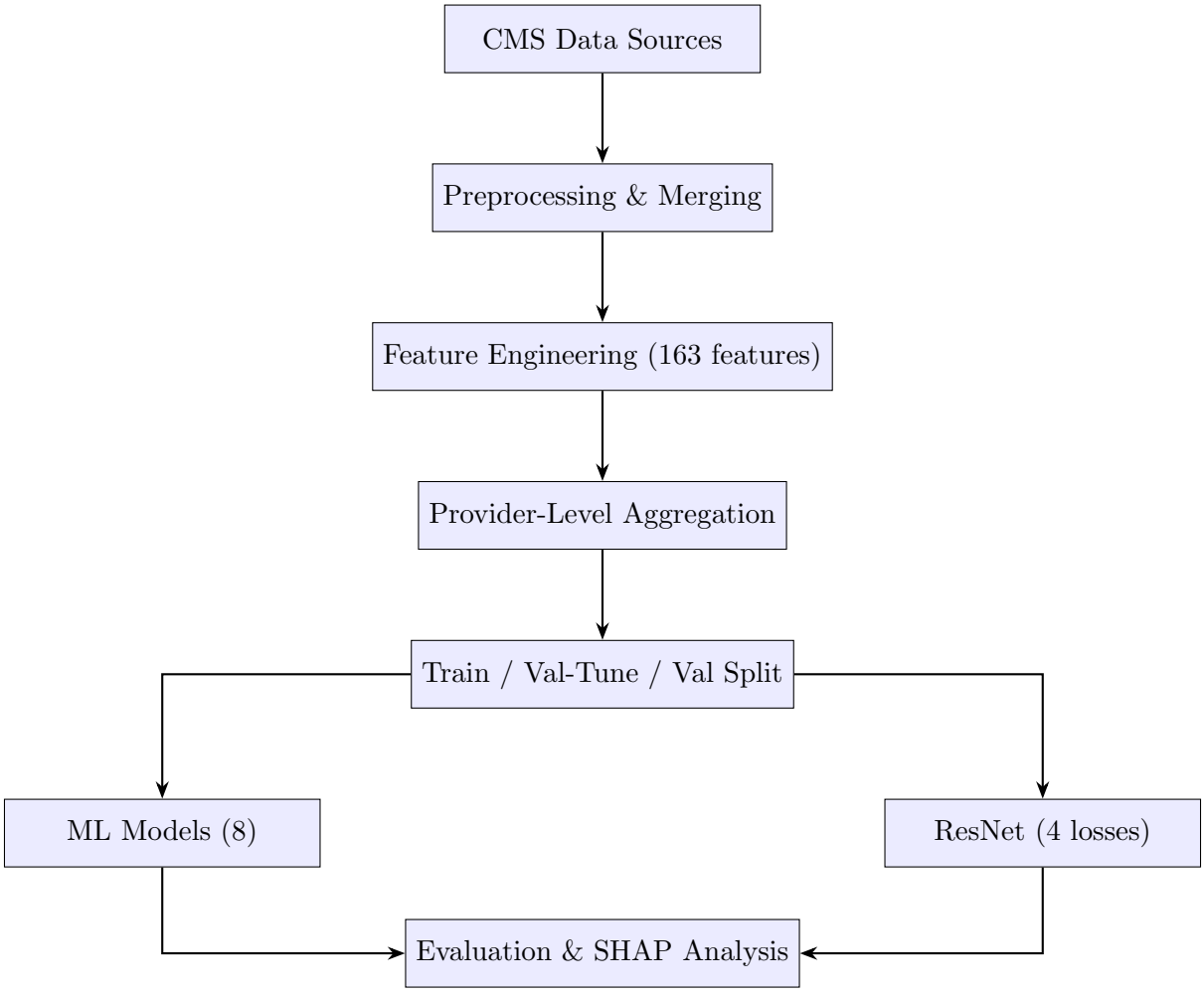


Figure 1: Proposed prediction framework for Medicare fraud detection

Figure 1 illustrates the proposed prediction framework for Medicare provider fraud detection. The pipeline begins with data ingestion and relational merging of four Kaggle CMS tables:

beneficiary demographics, inpatient claims, outpatient claims, and provider fraud labels. The merged data then undergoes cleaning and quality assurance, including duplicate removal, correction of data-type inconsistencies, and verification of provider identifiers. Missing values in claim-derived fields are handled through structural zero imputation, after which provider-level feature engineering produces 163 numerical variables through aggregation, one-hot encoding of categorical fields, and normalization of continuous features.

The resulting claim-level records are aggregated to the provider level, yielding one observation per provider. The dataset is then partitioned into three non-overlapping stratified subsets: a training set containing 49% of providers (2,650 observations), a tuning validation set containing 21% (1,137 providers), and a held-out evaluation set containing 30% (1,623 providers). The tuning validation set is used for early stopping in CatBoost and the ResNet loss function variants, while the held-out evaluation set is reserved exclusively for final performance reporting. Feature standardization is fit only on the training data and then applied unchanged to the tuning and evaluation sets to prevent data leakage. Optional SMOTE resampling is applied only to the training partition. All random operations use a fixed seed for reproducibility.

This design is consistent with the provider-aware validation concerns raised by Johnson and Khoshgoftaar (2023), who showed that conventional cross-validation can yield overly optimistic estimates when provider-level information leaks across folds. Hancock et al. (2024) similarly demonstrated that repeated stratified evaluation reduces instability caused by random partitioning. In the present thesis, eight traditional or ensemble machine learning models and four loss variants are evaluated within a common supervised benchmarking framework. Final evaluation is conducted on the held-out evaluation set using G-mean, accuracy, sensitivity, specificity, Cohen's kappa, and AUC, with SHAP analysis used to support interpretability.

3.2 Neural Network Architecture

The deep learning component of this thesis uses a tabular ResNet architecture following Gorishniy et al. (2021), implemented through the `rtdl_revisiting_models` library. Unlike standard feedforward networks, the ResNet uses residual blocks with skip connections that allow gradient flow across layers, improving training stability and enabling deeper architectures without degradation. This design suits tabular fraud data, where feature interactions may be complex, but the dataset size remains moderate.

Each residual block consists of a linear layer, batch normalization, ReLU activation, dropout at rate p_1, a second linear layer, and a skip connection back to the block input. The block width is controlled by d_{block}, and the hidden dimension within each block is determined by the product $d_{\text{block}} \times d_{\text{hidden_multiplier}}$. The number of residual blocks n_{blocks} controls network depth. An optional second dropout rate p_2 is applied after each block. The network terminates with a two-unit output layer that produces raw logits for binary classification.

In shorthand, each block follows the sequence Linear → BatchNorm → ReLU → Dropout(p_1) → Linear + skip. All weight matrices are initialized using He initialization (He et al., 2015). Training uses the Adam optimizer (Kingma and Ba, 2015) with a weight decay of 1e−4 and a `ReduceLROnPlateau` learning rate scheduler that halves the learning rate after 5 epochs of no improvement. Models are trained for up to 150 epochs with early stopping when the validation G-mean does not improve for 10 consecutive epochs.

The ResNet is evaluated under four loss-function variants: PF-Loss (PF), Weighted Cross-entropy (WCE), Focal Loss (FL), and Weighted LDAM (WLDAM). The network architecture remains fixed across loss functions, allowing the loss function to serve as the primary mechanism for comparing class-imbalance handling strategies in deep learning. The architecture search varies block width (d_{block}), hidden multiplier ($d_{\text{hidden_multiplier}}$), number of blocks ($n_{\text{blocks}}$), dropout rates, learning rates, and batch sizes across three candidate configurations.

3.3 Loss Functions

Parameter-Free Loss (PF-Loss)

Following Du et al. (2023), PF-Loss is formulated over each mini-batch by replacing individual correct-class probabilities in the standard cross-entropy objective with class-level average probabilities. Standard cross-entropy loss over a mini-batch $\mathcal{B}$ is given by

$$L_{\text{CE}} = -\frac{1}{|\mathcal{B}|} \sum_{i \epsilon \mathcal{B}} \log p(y_i \mid x_i) \tag{2}$$

where $p(y_i \mid x_i)$ is the predicted probability of the true label y_i for observation x_i, and $|\mathcal{B}|$ is the mini-batch size. In imbalanced settings, this loss is dominated by the majority class because the sum is taken over all observations in the batch.

PF-Loss addresses this issue by replacing each individual sample probability with the average predicted probability of its class within the current mini-batch. For a classification problem with M classes, PF-Loss is defined as

$$L_{\text{PF}} = -\frac{1}{M} \sum_{j=1}^{M} \log(q_j) \tag{3}$$

where

$$q_j = \frac{1}{N_j} \sum_{i=1}^{N_j} p_i \tag{4}$$

and

$$p_i = \sum_{j=1}^{M} y_{i,j}\, p_{i,j}. \tag{5}$$

Here, $p_{i,j}$ is the predicted probability that sample i belongs to class j, $y_{i,j}$ is the one-hot class indicator, p_i is the predicted probability assigned to the true class of sample i, N_j is the number of class j samples in the mini-batch, and M is the total number of classes. Thus, q_j represents the average correct-class probability for class j within the mini-batch.

The main difference between PF-Loss and standard cross-entropy is that PF-Loss averages over classes rather than over samples. As a result, each class contributes equally to the loss, regardless of its frequency in the batch. When the minority fraud class has a low average predicted probability, the loss increases and produces stronger corrective gradients for that class. This re-balancing occurs without requiring class weights, focusing parameters, or margin parameters.

PF-Loss has three practical advantages. First, it is parameter-free with respect to class imbalance. Second, it adapts naturally to the class composition of each mini-batch. Third, it reduces training and tuning time by eliminating loss-specific hyperparameter search. A complete derivation is provided in Appendix B.

Table 2: Variables used in the PF-Loss formulation

Symbol	**Definition**
x_i	Feature vector for provider i
y_i	True binary fraud label for provider i
$p(c \mid x_i)$	Predicted probability of class c for provider i
q_j	Average predicted probability for class j in the mini-batch
N_j	Number of samples from class j in the mini-batch
M	Total number of classes
L_{PF}	Parameter-Free Loss value
$\mathcal{B}$	Current mini-batch; $\lvert\mathcal{B}\rvert$ is the batch size

Gradient Analysis

For binary classification, PF-Loss can be written as

$$\mathrm{PF}_b = -\log\left(\frac{1}{N_1}\sum_i t_i y_i\right) - \log\left(\frac{1}{N_0}\sum_i (1-t_i)(1-y_i)\right) \tag{6}$$

where N_1 denotes the number of minority positive samples, N_0 denotes the number of majority negative samples, $t_i \in \{0, 1\}$ is the class indicator, and y_i is the estimated positive-class probability for sample i.

The gradient of PF-Loss with respect to the estimated output y_i is

$$\frac{\partial \mathrm{PF}_b}{\partial y_i} = -\frac{t_i y_i (1 - y_i)}{\sum_i t_i y_i} + \frac{(1 - t_i) y_i (1 - y_i)}{\sum_i (1 - t_i)(1 - y_i)} \tag{7}$$

This gradient differs from that of Focal Loss because the update for an observation depends on overall class-level performance rather than only on that sample's difficulty. Even when a minority-class instance is classified correctly, PF-Loss continues to provide corrective updates if the average minority-class performance remains weak.

Robustness to Outliers

PF-Loss uses the arithmetic mean of sample probabilities within each class, whereas weighted cross-entropy with standard class weights behaves more like a geometric aggregation. The arithmetic mean is less sensitive to extreme observations, so a single very low predicted probability has a smaller effect on PF-Loss than it would under a geometric-style aggregation. This property allows PF-Loss to emphasize class-level improvement without being dominated by isolated outliers.

Weighted Cross-Entropy (WCE) assigns greater weight to underrepresented classes:

$$\mathrm{WCE} = -\sum_{i=1}^{N} \alpha_i \log(p_i) \tag{8}$$

where α_i is the class weight and p_i is the predicted probability of the true class. Weights are typically set using inverse class frequency (Huang et al., 2016; Cui et al., 2019).

Focal Loss (FL) (Lin et al., 2017) reduces the influence of well-classified observations through a modulating factor:

$$\mathrm{FL} = -\sum_{i=1}^{N} \alpha_i (1 - p_i)^{\gamma} \log(p_i) \tag{9}$$

The focusing parameter γ and the class-balancing parameter α introduce additional tuning costs. Singh et al. (2023) proposed Batch-Balanced Focal Loss, combining balanced mini-batch sampling with focal loss.

Weighted Label-Distribution-Aware Margin (WLDAM) Loss (Cao et al., 2019) combines class-specific decision margins with a deferred reweighting schedule (DRW):

$$\mathcal{L}_{\mathrm{WLDAM}}((\mathbf{x}, y); f) = -w_y \log \frac{e^{z_y - \Delta_y}}{e^{z_y - \Delta_y} + \sum_{j \neq y} e^{z_j}} \tag{10}$$

where $\Delta_j = C/n_j^{1/4}$ and w_y are class-dependent weights derived from inverse class frequency. Kato and Hotta (2023) proposed Enlarged Large Margin (ELM) loss, which increases the true-class margin and penalizes the closest competing class.

Table 3: Loss functions used in neural network benchmarking

Loss Function	**Formula / Key Idea**	**Hyperparam.**
WCE (ResNet)	$\mathcal{L} = -\sum_i w_{y_i}\, y_i \log(p_i)$	α_{scale}
FL (ResNet)	$\mathcal{L} = -\sum_i \alpha\,(1-p_i)^{\gamma} \log(p_i)$	γ, α
WLDAM (ResNet)	$\mathcal{L} = -w_y \log \frac{e^{z_y - \Delta_y}}{e^{z_y - \Delta_y} + \sum_{j \neq y} e^{z_j}}$, $\Delta_j = C/n_j^{1/4}$	C_{wldam}, α_{scale}
PF (ResNet)	$\mathcal{L} = -(1/M) \sum_{j=1}^{M} \log(\bar{p}_j)$	None

3.4 Model Architectures

The modeling framework consists of three architectural groups: traditional and ensemble machine learning models, tabular ResNet neural network variants with different loss functions, and unsupervised autoencoder models for anomaly detection. The primary supervised benchmark includes eight traditional or ensemble machine learning models and four loss-function variants evaluated on a shared ResNet backbone. Autoencoder models are evaluated separately as unsupervised anomaly-detection baselines (all 4 tables with the best respective hyperparameters will be provided towards the bottom of this section).

Traditional and Ensemble Models

The supervised benchmark includes Regularized Logistic Regression (Reg. LR), Random Forest (RF), XGBoost (XGB), Gradient Boosting (GB), AdaBoost (AB), Decision Tree (DT), Support Vector Machine (SVM, Lin. & RBF), and CatBoost (CB). Regularized Logistic Regression is implemented with ℓ_2 regularization and built-in cross-validation for the regularization parameter C. Random Forest is tuned over the number of estimators, maximum depth, minimum samples split, and minimum samples leaf. XGBoost is tuned over learning rate, maximum depth, number of estimators, subsample ratio, and column sampling ratio. Gradient Boosting is tuned over learning rate, number of estimators, maximum depth, and subsample ratio, while AdaBoost is tuned over the number of estimators and learning rate. Decision Tree is tuned over maximum depth, minimum samples split, minimum samples leaf, and splitting criterion. SVM is evaluated with both linear and radial basis function kernels, with tuning over C and γ. CatBoost is tuned over depth, learning rate, ℓ_2 regularization, and the number of iterations.

Class imbalance handling is selected according to the capabilities of each implementation. Models that support native weighting use `class_weight='balanced'` as a baseline strategy.

XGBoost uses `scale_pos_weight`. Gradient Boosting and AdaBoost are implemented through `ImbPipeline` with StandardScaler, SMOTE, and the classifier because they do not directly support class weighting in the same way. CatBoost uses explicit class weights.

Unsupervised Autoencoder Models

Two unsupervised autoencoder models are included as anomaly-detection baselines. These models are trained exclusively on non-fraud observations so that the reconstruction error can be used as an indicator of anomalous provider behavior. The first model is a one-layer autoencoder with 15 encoding dimensions, and the second is a two-layer autoencoder with encoding dimensions of 14 and 7. In both cases, fraud risk is inferred by comparing the reconstruction error against an optimized threshold.

3.5 Hyperparameter Tuning

Hyperparameter tuning for traditional machine learning models is performed using 5-fold stratified cross-validation within `GridSearchCV`, with G-mean used as the model selection criterion. Grid search is adopted because the traditional model search spaces are relatively small, ranging from 5 to 90 configurations, so exhaustive evaluation remains practical. The number of configurations searched ranges from 5 for Regularized Logistic Regression, implemented through `LogisticRegressionCV`, to 90 for the Decision Tree. CatBoost is tuned separately using a manual grid of 27 configurations with early stopping on the tuning validation set. SAMME, which stands for Stagewise Additive Modeling using a Multiclass Exponential loss function, was introduced by Zhu et al. (2009) as an extension of the original AdaBoost algorithm for multiclass classification.

For the ResNet loss function variants, hyperparameter tuning is conducted through a structured architecture search, with G-mean on the tuning validation set used to determine the best configuration for each loss function. The base architecture grid contains three candidate configurations, varying block width, hidden multiplier, number of blocks, dropout rates, learning rates, and batch sizes. PF-Loss introduces no loss-specific hyperparameters, so its search remains limited to three configurations. By contrast, WCE-Loss expands the search to 12 configurations by adding four α_{scale} values, Focal Loss expands to 27 configurations by adding three γ and three α_{scale} values, and WLDAM expands to 36 configurations by adding four C and three α_{scale} values. This reduction in search dimensionality is among the main computational advantages of PF-Loss. Early stopping is applied to the tuning validation set, and final evaluation is conducted on the held-out evaluation set.

Table 4: Search space for hyperparameter tuning: traditional and ensemble models

Model	**Hyperparameter**	**Search Range**
Reg. LR	Inverse regularization strength (C_{lr})	Built-in CV
RF	Number of trees (`n_estimators`)	{200, 400, 600}
	Maximum tree depth (`max_depth`)	{4, 6, 8, None}
	Minimum samples to split (`min_samples_split`)	{2, 5, 10}
XGB	Number of boosting rounds (`n_estimators`)	{100, 200, 400}
	Maximum tree depth (`max_depth`)	{3, 5, 7}
	Step size shrinkage (`learning_rate`)	{0.01, 0.05, 0.1}
	Positive class weight (`scale_pos_weight`)	$\{n_-/n_+\}$
	Subsample ratio (`subsample`)	{0.8, 1.0}
GB	Number of boosting rounds (`n_estimators`)	{100, 200, 400}
	Maximum tree depth (`max_depth`)	{3, 5, 7}
	Step size shrinkage (`learning_rate`)	{0.01, 0.05, 0.1}
	Subsample ratio (`subsample`)	{0.8, 1.0}
AB	Number of estimators (`n_estimators`)	{50, 100, 200, 400}
	Step size shrinkage (`learning_rate`)	{0.01, 0.05, 0.1, 0.5, 1.0}
	Boosting algorithm (`algorithm`)	{SAMME, SAMME.R}
DT	Maximum tree depth (`max_depth`)	{3, 5, 8, 10, None}
	Minimum samples to split (`min_samples_split`)	{2, 5, 10}
	Minimum samples per leaf (`min_samples_leaf`)	{1, 2, 5}
	Class weight (`class_weight`)	{balanced}
	Split criterion (`criterion`)	{gini, entropy}
SVM (RBF)	Penalty parameter ($C_{svm,r}$)	$\{0.1, 1, 10\}$
	Kernel coefficient (`gamma`)	$\{\text{scale}, 0.01, 0.001\}$
SVM (Lin.)	Penalty parameter ($C_{svm,l}$)	$\{0.01, 0.1, 1, 10\}$
CB	Maximum tree depth (`depth`)	{4, 6, 8}
	Step size shrinkage (`learning_rate`)	{0.03, 0.05, 0.1}
	L2 regularization (`l2_leaf_reg`)	{1, 3, 5}

Table 5: Search space for hyperparameter tuning: ResNet loss function variants

Model	**Hyperparameter**	**Search Range**
NN Arch.	Residual blocks (`n_blocks`)	{1, 2, 3}
	Block dimension (`d_block`)	{64, 128}
	Hidden multiplier (`d_hidden_multiplier`)	{2.0, 4.0}
	Dropout rate (p_1)	{0.15, 0.25}
	Learning rate (`lr`)	{0.0005, 0.001}
	Batch size	{32, 64}
	Hidden-layer activation	ReLU (fixed)
	Output-layer activation	Softmax (fixed)
	Optimization method	Adam (fixed)
PF (ResNet)	(none)	Parameter-free
WCE (ResNet)	Class weight scale (α_{scale})	{0.5, 1.0, 1.5, 2.0}
FL (ResNet)	Focusing parameter (γ)	{1.0, 2.0, 3.0}
	Class weight scale (α_{scale})	{0.5, 1.0, 2.0}
WLDAM (ResNet)	Margin scaling constant (C_{wldam})	{0.1, 0.3, 0.5, 1.0}
	Class weight scale (α_{scale})	{0.5, 1.0, 2.0}

Table 6: Best hyperparameters by grid search: traditional and ensemble models

Model	**Hyperparameter**	**Best Value**
Reg. LR	Inverse regularization strength (C_{lr})	0.0001
RF	Number of trees (`n_estimators`)	600
	Maximum tree depth (`max_depth`)	4
	Minimum samples to split (`min_samples_split`)	2
	Class weight (`class_weight`)	balanced
XGB	Number of boosting rounds (`n_estimators`)	100
	Maximum tree depth (`max_depth`)	3
	Step size shrinkage (`learning_rate`)	0.01
	Positive class weight (`scale_pos_weight`)	9.69
	Subsample ratio (`subsample`)	1.0
GB	Number of boosting rounds (`n_estimators`)	100
	Maximum tree depth (`max_depth`)	3
	Step size shrinkage (`learning_rate`)	0.01
	Subsample ratio (`subsample`)	1.0
AB	Number of estimators (`n_estimators`)	100
	Step size shrinkage (`learning_rate`)	0.05
	Boosting algorithm (`algorithm`)	SAMME
DT	Maximum tree depth (`max_depth`)	3
	Minimum samples to split (`min_samples_split`)	2
	Minimum samples per leaf (`min_samples_leaf`)	1
	Class weight (`class_weight`)	balanced
	Split criterion (`criterion`)	gini
SVM (RBF)	Penalty parameter ($C_{svm,r}$)	10
	Kernel coefficient (`gamma`)	0.001
SVM (Lin.)	Penalty parameter ($C_{svm,l}$)	0.01
CB	Maximum tree depth (`depth`)	4
	Step size shrinkage (`learning_rate`)	0.05
	L2 regularization (`l2_leaf_reg`)	5

Table 7: Best hyperparameters by grid search: ResNet loss function variants[a]

Model	Hyperparameter	Best Value
PF (ResNet)	Residual blocks (`n_blocks`)	1
	Total linear layers	4
	Hidden multiplier (`d_hidden_multiplier`)	2.0
	Neurons per block	$64 \to 128 \to 64$
	Learning rate (`lr`)	0.001
	Batch size	64
WCE (ResNet)	Residual blocks (`n_blocks`)	2
	Total linear layers	6
	Hidden multiplier (`d_hidden_multiplier`)	2.0
	Neurons per block	$128 \to 256 \to 128$
	Learning rate (`lr`)	0.001
	Batch size	64
	Class weight scale (α_{scale})	1.0
FL (ResNet)	Residual blocks (`n_blocks`)	1
	Total linear layers	4
	Hidden multiplier (`d_hidden_multiplier`)	2.0
	Neurons per block	$64 \to 128 \to 64$
	Learning rate (`lr`)	0.001
	Batch size	64
	Focusing parameter (γ)	3.0
	Class weight scale (α_{scale})	1.0
	Residual blocks (`n_blocks`)	1
	Total linear layers	4

Continued on next page

Table 7 (continued)

Model	Hyperparameter	Best Value
	Hidden multiplier (`d_hidden_multiplier`)	2.0
WLDAM (ResNet)	Neurons per block	$64 \to 128 \to 64$
	Learning rate (`lr`)	0.001
	Batch size	64
	Class weight scale (α_{scale})	1.0
	Margin scaling constant (C_{wldam})	0.5

[a]All variants share the same training setup: ReLU hidden activation, Softmax output, Adam optimizer with weight decay 1e−4, ReduceLROnPlateau scheduler, and early stopping with patience = 10.

3.6 Evaluation Metrics

All models are evaluated using standard classification metrics derived from the confusion matrix shown in Table 8. For binary classification, the confusion matrix consists of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

Table 8: Confusion matrix for binary classification

	Actual Positive	**Actual Negative**
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Using this notation, the evaluation metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$\text{Sensitivity} = \text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{13}$$

$$\text{G-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{14}$$

$$\text{AUC} = \text{Area under the ROC curve} \tag{15}$$

G-mean is used as the primary evaluation metric because it captures the balance between sensitivity and specificity without being dominated by majority-class performance. In imbalanced fraud detection, a model may achieve high accuracy by predicting the majority non-fraud class for most observations while still failing to identify fraudulent providers. In such cases, G-mean provides a more informative summary of performance than accuracy alone.

3.7 Experimental Design

The Kaggle CMS dataset contains 5,410 providers, of whom 506 are labeled as 'Fraud' and 4,904 are labeled as 'Non-Fraud'. The primary supervised benchmark evaluates twelve models: eight traditional or ensemble machine learning classifiers and four ResNet loss-function variants. Hyperparameters are selected through the tuning strategies described above, and all final results are reported on the held-out evaluation set. In a secondary analysis, class-imbalance strategies such as SMOTE are varied while holding the best model hyperparameters fixed.

3.8 Quantitative Risk Analysis

Beyond binary classification, this thesis extends fraud detection outputs into a quantitative risk framework that converts predicted probabilities into dollar-denominated measures of risk. Drawing on Basel II/III credit risk methodology, the framework includes four components: probability calibration, expected loss estimation, budget-constrained audit optimization, and Monte Carlo simulation.

Probability Calibration

Probability calibration is evaluated using the Brier Score, defined as the mean squared error between predicted probabilities and observed outcomes. Reliability diagrams are also used to compare predicted probabilities with observed fraud frequencies across bins. For downstream risk estimation, the best-calibrated model is selected because accurate probability estimates are more important than binary classification accuracy when outputs are translated into dollar-denominated risk measures.

Expected Loss Estimation

Expected loss for provider i is defined as

$$EL_i = P_i \times \text{Exposure}_i \tag{16}$$

where P_i is the predicted fraud probability and Exposure_i is the provider's total reimbursement amount. Risk-decile analysis is used to examine the concentration of expected loss across providers.

Budget-Constrained Audit Optimization

Since fraud investigation units face limited budgets, audit selection is modeled as a 0–1 knapsack problem solved through linear programming using `scipy.optimize.linprog`. The objective is to maximize the expected loss captured under a fixed budget. Audit cost per provider is modeled as a \$5,000 base cost plus \$50 per claim.

Monte Carlo Simulation

In quantifying the uncertainty of the expected loss estimates, 10,000 Monte Carlo simulations are conducted. In each simulation, provider fraud status is sampled from a Bernoulli distribution with probability P_i, and recovery rates are sampled from a Beta(8, 6) distribution with a mean of approximately 57%. The resulting loss distribution is used to estimate Value-at-Risk (VaR), Expected Shortfall, and confidence intervals for total system loss.

Table 9: Quantitative risk measures and applications

Risk Measure	**Description**	**Application**
Expected Loss (EL)	$P(\text{fraud}) \times \text{Exposure}$	Provider risk ranking
VaR (95%)	95th percentile of loss distribution	Budget planning
VaR (99%)	99th percentile of loss distribution	Worst-case planning
CVaR / ES	Mean loss beyond VaR threshold	Tail risk quantification
Gini Coefficient	Concentration of expected loss	Resource allocation efficiency
Efficient Frontier	Maximum EL captured by budget	Audit optimization

4 Data Description

4.1 Data Sources

The data used in this study are publicly available through the Healthcare Provider Fraud Detection dataset on Kaggle [60]. Although alternative CMS data pipelines exist, including those of Johnson and Khoshgoftaar [4] and Hancock et al.[13], the Kaggle dataset is used here because of its benchmark for model testing and clearly defined provider-level fraud labels. The dataset consists of eight '.csv' files organized into training and testing subsets.

Table 10: CMS dataset files and their dimensions

File	Records	Features	Description
`Train.csv`	5,410	2	Provider identifiers with fraud labels
`Train_Beneficiarydata.csv`	138,556	25	Beneficiary demographics and chronic conditions
`Train_Inpatientdata.csv`	40,474	30	Inpatient claims, diagnoses, and procedures
`Train_Outpatientdata.csv`	517,737	27	Outpatient claims and financial amounts
`Test.csv`	1,353	1	Provider identifiers without labels
`Test_Beneficiarydata.csv`	63,968	25	Same structure as training beneficiary data
`Test_Inpatientdata.csv`	9,551	30	Same structure as training inpatient data
`Test_Outpatientdata.csv`	125,841	27	Same structure as training outpatient data

The training tables include beneficiary demographics (date of birth (`DOB`), date of death (`DOD`), gender, race, chronic condition indicators, Medicare coverage months for Part A (`NoOfMonths_PartACov`) and Part B (`NoOfMonths_PartBCov`), and reimbursement and deductible amounts), inpatient claims (claim dates, physician identifiers, up to ten ICD diagnosis code fields (`ClmDiagnosisCode_1` through `ClmDiagnosisCode_10`) and up to six procedure code fields (`ClmProcedureCode_1` through `ClmProcedureCode_6`), the insurance claim amount reimbursed (`InscClaimAmtReimbursed`), the deductible amount paid (`DeductibleAmtPaid`), and admission and discharge dates), outpatient claims (similar structure without admission and discharge dates), and provider-level fraud labels coded as `PotentialFraud` (Yes/No). Following Hancock et al. [13], provider names and addresses are excluded to prevent model memorization, which occurs when learning models reproduce specific verbatim training data that is unnecessary for research.

4.2 Data Merging and Feature Engineering

A three-stage merging procedure creates the modeling dataset: (1) inpatient and outpatient claims are combined through an outer join on 28 shared columns, including the beneficiary identifier (`BeneID`), claim identifier (`ClaimID`), provider identifier (`Provider`), claim dates, physician identifiers, diagnosis codes, procedure codes, and reimbursement and deductible amounts; (2) beneficiary demographics are merged through an inner join on `BeneID`; (3) provider-level fraud labels are joined through `Provider`. The resulting dataset contains 558,211 claim-level observations and 58 variables. A total of 163 provider-level features are then engineered through aggregation, following the data-enrichment strategy of Johnson and Khoshgoftaar [4].

Table 11: Feature categories and descriptions

Category	**Example Features**	**Count**
Claim-Level Aggregations	Provider-level averages of `InscClaimAmtReimbursed`, `DeductibleAmtPaid`, `IPAnnualReimbursementAmt`, `OPAnnualReimbursementAmt`, and `AdmitForDays`; claim counts by provider, beneficiary, and physician	73
Beneficiary Demographics	Age, gender, chronic condition indicators (11 binary flags), Medicare Part A/B coverage months, `WhetherDead`, `RenalDiseaseIndicator`	31
Diagnosis Code Features	Averages of financial variables grouped by primary through quaternary diagnosis codes (`ClmDiagnosisCode_1` through `ClmDiagnosisCode_4`), admission diagnosis code (`ClmAdmitDiagnosisCode`), and diagnosis-related group code (`DiagnosisGroupCode`)	22
Temporal Features	`AdmitForDays` (length of hospital stay, computed as discharge date minus admission date plus one), averages of `AdmitForDays` grouped by diagnosis and procedure codes	18
Provider-Beneficiary Relations	Claim counts across provider-beneficiary (`ClmCount_Provider_BeneID`), provider-physician, and provider-procedure code pairings	19
Total		**163**

Claim-level aggregations summarize financial and utilization variables at the provider level. For each provider, mean values of `InscClaimAmtReimbursed` (the dollar amount Medicare reimbursed per claim), `DeductibleAmtPaid` (the deductible paid by the beneficiary per claim), `IPAnnualReimbursementAmt` (annual inpatient reimbursement per beneficiary),

IPAnnualDeductibleAmt (annual inpatient deductible), OPAnnualReimbursementAmt (annual outpatient reimbursement per beneficiary), OPAnnualDeductibleAmt (annual outpatient deductible), and AdmitForDays (length of stay in days) are computed, along with analogous averages grouped by beneficiary (e.g., PerBeneIDAvg_InscClaimAmtReimbursed) and by each physician role (e.g., PerAttendingPhysicianAvg_*). Claim count features tally the number of claims per provider overall (ClmCount_Provider) and per provider-entity pairing, such as provider-beneficiary (ClmCount_Provider_BeneID) and provider-attending physician (ClmCount_Provider_AttendingPhysician).

Beneficiary demographics aggregate patient characteristics, including 11 chronic condition indicators (e.g., ChronicCond_Heartfailure, ChronicCond_Diabetes, ChronicCond_Alzheimer), which are recoded from their original CMS values of 1 (condition present) and 2 (condition absent) to binary 1/0 flags before aggregation. The RenalDiseaseIndicator, originally coded as 'Y' or '0' in the raw data, is similarly converted to a binary 1/0 indicator. Diagnosis code features capture diagnostic diversity by computing average financial and utilization metrics grouped by each of the first four ICD diagnosis code fields, the admission diagnosis code (ClmAdmitDiagnosisCode, the condition identified at hospital admission), and the diagnosis-related group code (DiagnosisGroupCode, a CMS classification that groups inpatient stays into clinically similar categories for reimbursement). Temporal features characterize claim timing, including AdmitForDays, which is computed as the number of days from the admission date (AdmissionDt) to the discharge date (DischargeDt) plus one. Provider-beneficiary relationship features capture structural patterns through claim counts across provider-beneficiary, provider-physician, and provider-procedure code pairings. After one-hot encoding categorical variables (gender, race, and binned Medicare Part A/B coverage months) with drop_first=True and Min-Max scaling of continuous features, the final dataset contains 163 features.

4.3 Missing Data Analysis

Missing data are predominantly structural rather than random. Among 558,211 claims, 92.75% are outpatient and therefore lack admission or discharge dates. Procedure code fields follow a decreasing pattern (4.18% populated for the first field, declining to 0% by the sixth). 'DOD' (date of death) is missing for 99.26% of records (living beneficiaries), and 'OperatingPhysician' is missing in 79.50% (outpatient claims without operating physicians). Hancock et al. [13] note that CMS data suppression for small cell sizes contributes to additional structured missingness. Missing values are handled through structural zero imputation for claim-derived fields, mode imputation for chronic condition indicators (3-4% missing), deriving a binary 'WhetherDead' indicator from DOD (coded as 1 if DOD is present and 0 otherwise), and dropping high-missingness physician identifier columns. Overall, these values are missing by design in the data, as many tabular columns are blank and not applicable to the patient.

Table 12: Missing data analysis for key features

Feature	Missing N	Missing %	Type	Handling
ClmProcedureCode_6	558,211	100.00	Structural	Impute 0
DOD	554,080	99.26	Structural	Use WhetherDead, drop
AdmitForDays	517,737	92.75	Structural	Impute 0 (outpatient)
OperatingPhysician	443,755	79.50	Structural	Drop column
ClmDiagnosisCode_4	393,616	70.52	Structural	Impute 0
DeductibleAmtPaid	899	0.16	Data quality	Impute 0

4.4 Data Partitioning

The 5,410 providers are partitioned into three stratified subsets, preserving the class distribution. The three-way split is necessary, given that CatBoost and loss functions rely on early stopping, which requires a separate tuning set to prevent leakage into the held-out evaluation set.

Table 13: Data partitioning strategy

Partition	Split	Providers	Non-Fraud	Fraud	Usage
Training Set	49%	2,650	2,402	248	Model fitting and CV
Tuning Set	21%	1,137	1,031	106	Early stopping only
Evaluation Set	30%	1,623	1,471	152	Final evaluation
Total	**100%**	**5,410**	**4,904**	**506**	

Providers = 'Non-Fraud' + 'Fraud'

4.5 Descriptive Statistics

The provider-level class distribution is highly imbalanced: 4,904 providers (90.65%) are Non-Fraud and 506 (9.35%) are Fraud. At the claim level, the distribution shifts to 61.88% Non-Fraud and 38.12% Fraud, indicating that fraudulent providers generate substantially more claims on average.

Table 14: Fraud versus non-fraud comparison of key characteristics

Characteristic	**Non-Fraud**	**Fraud**
Avg Claim Reimbursement (`InscClaimAmtReimbursed`, $)	$755.21	$1,389.51
Avg Claim Deductible (`DeductibleAmtPaid`, $)	$54.28	$117.65
Avg IP Annual Reimb. per Beneficiary (`IPAnnualReimbursementAmt`, $)	$4,500	$7,200
Total Unique Physicians per Provider	9.1	18.4
`DiagnosisCode_Entropy`	2.78	2.31
Ratio Inpatient to Outpatient	0.41	0.74
Avg Age (Years)	73.69	73.90
Heart Failure (`ChronicCond_Heartfailure`, %)	58.69%	59.61%
Diabetes (`ChronicCond_Diabetes`, %)	70.35%	70.85%

Fraudulent providers show markedly higher billing amounts, with nearly double the average claim reimbursement and around 60% higher average inpatient reimbursement per beneficiary—suggesting systematic upcoding. They also interact with roughly twice as many unique physicians (18.4 vs. 9.1), computed as the count of distinct attending, operating, and other physician identifiers associated with a given provider, possibly reflecting suspicious referral patterns. `DiagnosisCode_Entropy` (Shannon entropy of 3-digit ICD codes) is lower for fraudulent providers (2.31 vs. 2.78), consistent with billing behavior concentrated on a narrow set of high-reimbursement diagnoses. The inpatient-to-outpatient ratio is higher (0.74 vs. 0.41), computed as the number of inpatient claims divided by outpatient claims per provider, reflecting a tendency toward costlier inpatient services. In contrast, the prevalence of age and chronic conditions remains similar across groups, indicating that the distinguishing signals are tied to billing behavior rather than patient demographics. Exploratory visualizations confirm these patterns and reveal subtle separability between fraud and non-fraud clusters prior to model training.

4.6 Comparative Model Breakdown

For convenience throughout this thesis, the twelve supervised models are referred to by the following abbreviations: Regularized Logistic Regression (Reg. LR), Random Forest (RF), XGBoost (XGB), Gradient Boosting (GB), AdaBoost (AB), Decision Tree (DT), Support Vector Machine (SVM), CatBoost (CB), and four tabular ResNet variants distinguished by their loss function — ResNet with Parameter-Free Loss (PF), ResNet with Weighted Cross-Entropy (WCE), ResNet with Focal Loss (FL), and ResNet with Weighted LDAM (WLDAM). These abbreviations are used consistently in all tables and figures that follow.

4.7 Quantitative Risk Analysis

To contextualize the classification task in economic terms, each provider's financial exposure is computed directly from raw claim-level reimbursement data prior to any normalization. Financial exposure quantifies the total dollar amount at risk per provider:

$$\begin{aligned}\text{Exposure}_i = \sum_{\text{claims}} \big(&\texttt{InscClaimAmtReimbursed} \\ &+ \texttt{IPAnnualReimbursementAmt} \\ &+ \texttt{OPAnnualReimbursementAmt}\big)\end{aligned} \tag{17}$$

This measure aggregates claim-level reimbursements with beneficiary-level annual program amounts, capturing both direct billing and systemic program costs associated with each provider.

Table 15: Provider-level financial exposure summary statistics (USD)

Statistic	**Value**
Mean Exposure per Provider	$575,849
Standard Deviation	$1,150,486
Minimum	$20
25th Percentile	$68,075
Median	$211,660
75th Percentile	$556,548
Maximum	$22,342,540
Total System Exposure	$3,115,340,970
Fraud Provider Exposure	$1,218,068,760
Non-Fraud Provider Exposure	$1,897,272,210

The exposure distribution is heavily right-skewed, with a mean roughly 2.7 times the median, indicating that a small number of high-volume providers account for a disproportionate share of total system exposure. Across all 5,410 providers, the total exposure amounts to approximately $3.12 billion, of which $1.22 billion (39.1%) is attributable to the 506 providers labeled as potentially fraudulent — despite representing only 9.35% of the provider population.

Table 16: Financial exposure by fraud status

Group	**Providers**	**Median Exposure**	**Total Exposure**
Non-Fraud	4,904	$179,870	$1,897,272,210
Fraud	506	$1,493,825	$1,218,068,760
All Providers	**5,410**	**$211,660**	**$3,115,340,970**

The median exposure for fraudulent providers ($1,493,825) is more than eight times higher than that of non-fraud providers ($179,870), reinforcing the billing-pattern disparities observed in the descriptive statistics. This concentration of financial exposure among a small provider subset inspires the quantitative risk framework, where classification outputs are translated into dollar-expected losses, budget-constrained investigation strategies, and Monte Carlo tail-risk measures to support resource allocation decisions.

5 Computational Results

All models were trained on a training partition containing 2,650 providers and evaluated on a held-out validation set of 1,623 providers. The primary evaluation metric was G-mean, the geometric mean of sensitivity and specificity. G-mean is especially appropriate for imbalanced classification because it penalizes models that achieve strong performance on one class while failing on the other. In this dataset, approximately 90% of observations belonged to the non-fraud class. As a result, a naive classifier that predicts non-fraud for every provider would still obtain 90% accuracy; yet, it would achieve 0% sensitivity and a G-mean of 0. All analyses were conducted in Python using Google Colab, with a random seed of 123 or 42 for the selected models to ensure reproducibility.

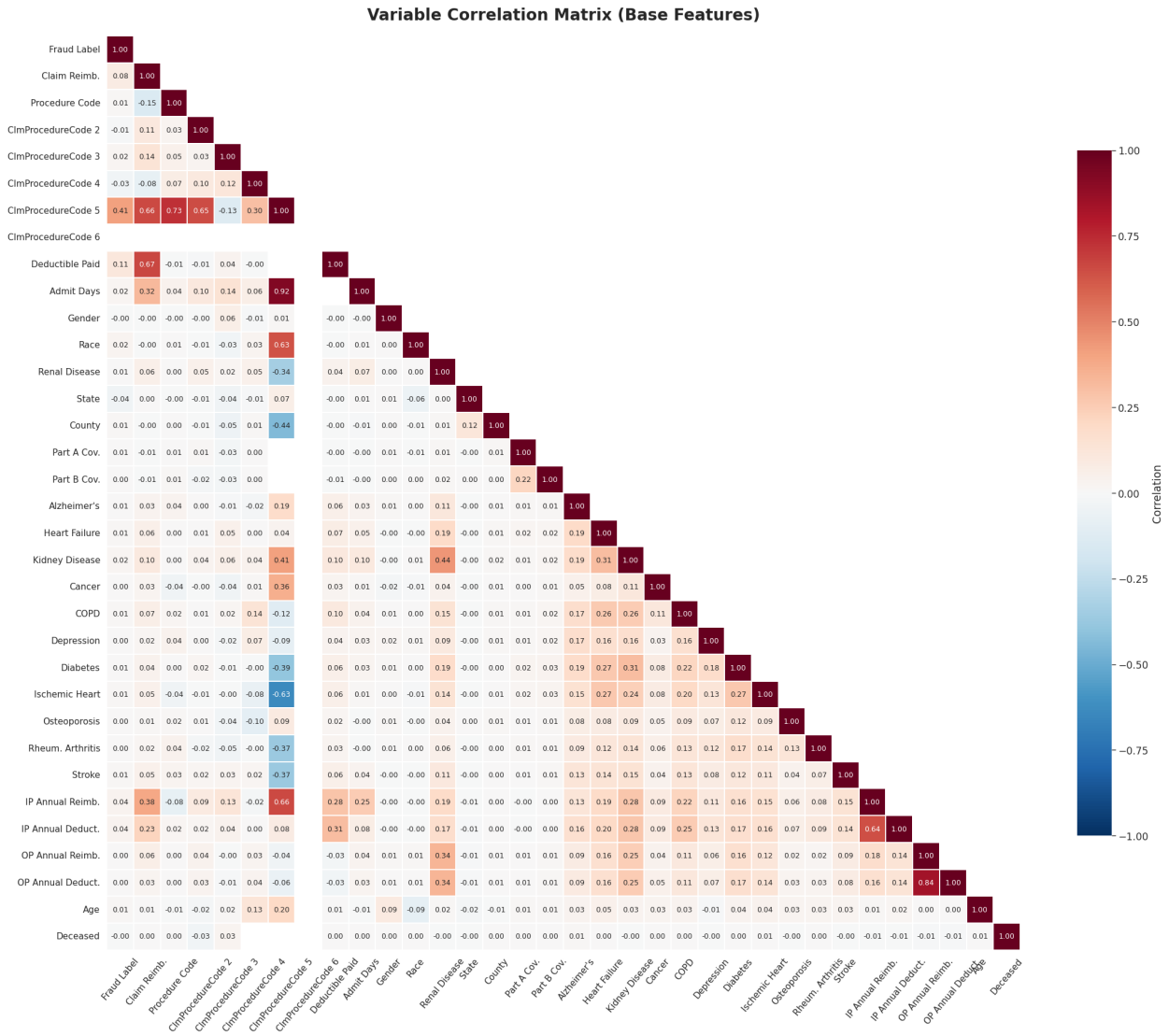


Figure 2: Variable correlation matrix

5.1 Cross-Validation Framework

Traditional machine learning models were tuned using 5-fold stratified cross-validation within `GridSearchCV`, with G-mean used as the primary scoring metric. The number of candidate configurations ranged from 5 for Regularized Logistic Regression to 90 for the Decision Tree. Since each configuration was evaluated across 5 folds, the tuning procedure required hundreds of model fits for each classifier. CatBoost was tuned separately using a manual grid of 27 configurations with early stopping on the tuning set. Regarding the neural networks, three architecture configurations were evaluated for each loss function, and additional loss-specific hyperparameters further expanded the overall search space.

5.2 Model Performance

Table 17 provides a comprehensive comparison of the performance of all twelve models on the same held-out validation set. Models are ranked by G-mean, with neural network variants grouped at the bottom for ease of interpretation.

Table 17: Benchmark model performance comparison

Model	Accuracy	Sensitivity	Specificity	Kappa	AUC	G-mean
XGB	0.8250	0.8882	0.8185	0.4067	0.9314	0.8526
AB	0.8213	0.8882	0.8144	0.4003	0.9284	0.8505
CB	0.8595	0.8355	0.8620	0.4574	0.9328	0.8487
DT	0.8053	**0.8947**	0.7961	0.3759	0.8993	0.8440
RF	0.8823	0.7961	0.8912	0.4979	0.9306	0.8423
GB	0.8534	0.8158	0.8572	0.4379	0.9288	0.8363
Reg. LR	0.9045	0.6513	0.9307	**0.5085**	0.9234	0.7786
SVM (RBF)	0.9211	0.4145	0.9735	0.4551	0.9261	0.6352
SVM (Lin.)	**0.9316**	0.3684	**0.9898**	0.4707	**0.9352**	0.6039
PF (ResNet)	0.8651	0.8684	0.8647	0.4798	0.9305	**0.8666**
FL (ResNet)	0.8688	0.8421	0.8715	0.4800	0.9344	0.8567
WCE (ResNet)	0.8404	0.8750	0.8368	0.4309	0.9328	0.8557
WLDAM (ResNet)	0.8466	0.8553	0.8457	0.4368	0.9341	0.8505

Several findings emerge from Table 17: Among the traditional/ensemble machine learning models, XGBoost produced the highest G-mean (0.8526), followed closely by AdaBoost (0.8505) and CatBoost (0.8487). Among the neural network loss variants, PF-Loss achieved the strongest G-mean (0.8666), with Focal Loss next at 0.8567 and WCE-Loss at 0.8557. All four ResNet variants outperformed every traditional model in G-mean, demonstrating the advantage of deep learning with imbalance-aware loss functions for this task. PF-Loss is particularly notable because it achieved the highest G-mean overall (0.8666) while requiring no additional hyperparameters for class-imbalance handling.

Regularized Logistic Regression and SVM achieved the highest specificity values: 0.9307 and 0.9898 (Linear kernel), respectively, but also the lowest sensitivity values: 0.6513 and 0.3684. This indicates that both models favored the majority class, yielding high accuracy at the expense of fraud detection performance. As a result, their lower G-mean values reinforce the accuracy as a sole metric for imbalanced data. In contrast, the Decision Tree produced the highest sensitivity (0.8947) among the traditional models, although its specificity was lower (0.7961). Among the neural network variants, WCE-Loss achieved the highest sensitivity (0.8750), while Focal Loss achieved the highest specificity (0.8715). The best AUC values were obtained by SVM Linear (0.9352), Focal Loss (0.9344), and WLDAM (0.9341), suggesting strong overall class separation.

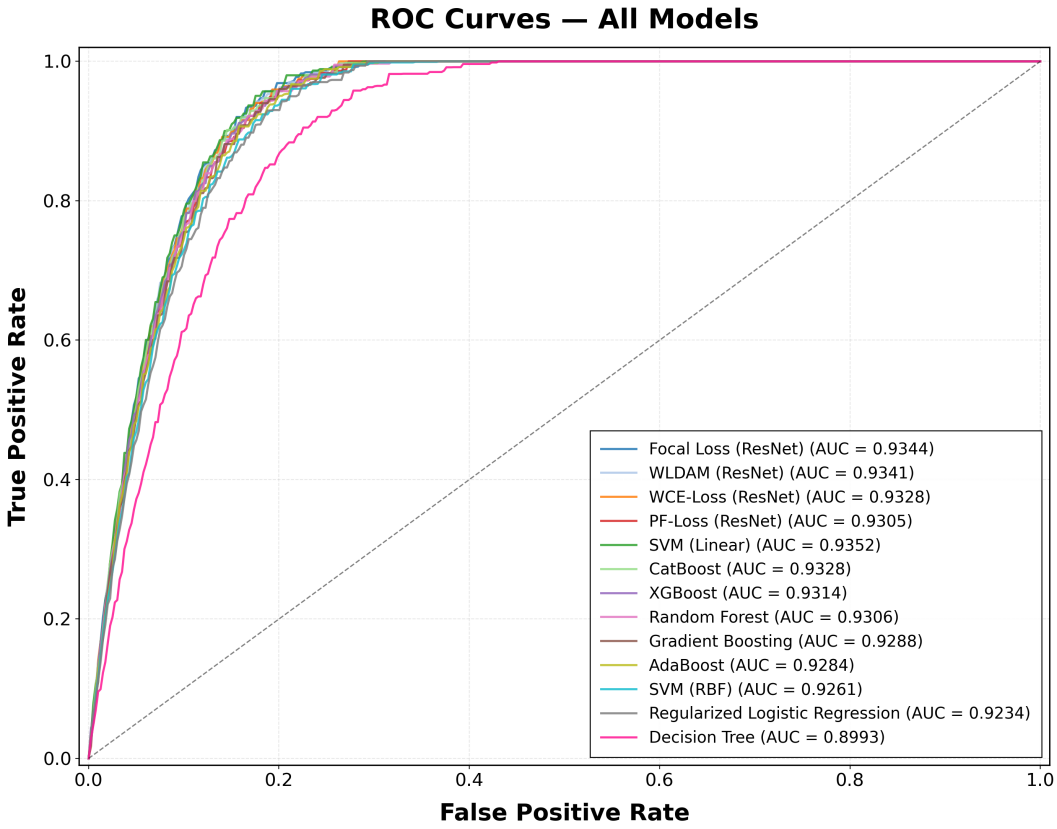


Figure 3: ROC Curves - All Models



Figure 4: Training Loss Curves - ResNet Models

Table 18: Nemenyi post-hoc test p-values for pairwise classifier comparisons

	AB	CB	DT	GB	RF	LR	FL	PF	WCE	WLDAM	SVM (Lin.)	SVM (RBF)	XGB
AB	1.00	1.00	0.89	1.00	1.00	0.12	1.00	1.00	1.00	1.00	1.00	0.66	0.99
CB	1.00	1.00	1.00	1.00	1.00	0.74	0.63	0.74	0.74	1.00	1.00	1.00	1.00
DT	0.89	1.00	1.00	1.00	1.00	0.99	0.18	0.26	0.26	0.96	1.00	1.00	1.00
GB	1.00	1.00	1.00	1.00	1.00	0.63	0.74	0.84	0.84	1.00	1.00	0.99	1.00
RF	1.00	1.00	1.00	1.00	1.00	0.48	0.86	0.92	0.92	1.00	1.00	0.96	1.00
Reg. LR	0.12	0.74	0.99	0.63	0.48	1.00	$\ll 0.01$	$\ll 0.01$	$\ll 0.01$	0.22	0.54	1.00	0.86
FL (ResNet)	1.00	0.63	0.18	0.74	0.86	$\ll 0.01$	1.00	1.00	1.00	0.98	0.81	0.06	0.48
PF (ResNet)	1.00	0.74	0.26	0.84	0.92	$\ll 0.01$	1.00	1.00	1.00	0.99	0.89	0.10	0.60
WCE (ResNet)	1.00	0.74	0.26	0.84	0.92	$\ll 0.01$	1.00	1.00	1.00	0.99	0.89	0.10	0.60
WLDAM (ResNet)	1.00	1.00	0.96	1.00	1.00	0.22	0.98	0.99	0.99	1.00	1.00	0.81	1.00
SVM (Lin.)	1.00	1.00	1.00	1.00	1.00	0.54	0.81	0.89	0.89	1.00	1.00	0.98	1.00
SVM (RBF)	0.66	1.00	1.00	0.99	0.96	1.00	0.06	0.10	0.10	0.81	0.98	1.00	1.00
XGB	0.99	1.00	1.00	1.00	1.00	0.86	0.48	0.60	0.60	1.00	1.00	1.00	1.00

5.3 PF-Loss vs. Baseline Losses

Table 19 presents a focused comparison of the four loss function variants, which were evaluated using the same ResNet architecture and training procedure. Under this setup, the loss function is the only factor that differs across models.

Table 19: Neural network loss function comparison

Loss Function	Best Architecture	Time
PF (ResNet)	d=64, 1 blk, dr=0.15, lr=1e-3	**19.7s**
FL (ResNet)	d=64, 1 blk, dr=0.15, lr=1e-3	189.1s
WCE (ResNet)	d=128, 2 blks, dr=0.25, lr=1e-3	73.1s
WLDAM (ResNet)	d=64, 1 blk, dr=0.15, lr=1e-3	227.2s

PF-Loss achieved the highest G-mean (0.8666) among the four loss functions, indicating the best overall balance between fraud detection and false alarm avoidance. Focal Loss ranked second in G-mean (0.8567) with the highest specificity among the neural network variants (0.8715). WCE-Loss achieved the highest sensitivity (0.8750) among the four, meaning it captured the largest share of true fraud cases, though at the cost of somewhat lower specificity. WLDAM produced a G-mean of 0.8505 with a balanced sensitivity–specificity profile (0.8553 and 0.8457). The most important practical advantage of PF-Loss, however, was computational efficiency. PF-Loss completed training in 19.7 seconds, compared with 73.1 seconds for WCE, 189.1 seconds for Focal Loss, and 227.2 seconds for WLDAM. This 3.7 to 11.5 times reduction in training time was driven by its smaller hyperparameter search space. Given that PF-Loss does not introduce loss-specific hyperparameters, only three architecture configurations were evaluated, whereas WCE required 12 configurations, Focal Loss required 27, and WLDAM required 36.

5.4 Class Imbalance Strategy Comparison

To evaluate the effect of different class imbalance handling strategies in a systematic way, each model was retrained under five conditions: (1) Original, with no resampling, (2) `ROS`, (3) `RUS`, (4) SMOTE, and (5) both `ROS` and `RUS`. For each condition, the model retained the best hyperparameters from the primary tuning phase, so the imbalance handling method was the only factor that varied. Table 9 presents the resulting G-mean scores, scaled by 100 for readability.

As for Table 20, the traditional machine learning models exhibited dramatically lower G-mean scores under the Original (no resampling) condition, with values ranging from 45.79 (XGBoost) to 69.47 (Decision Tree). Therefore, without explicit class imbalance handling, most traditional classifiers default to predicting the majority class. In contrast, the ResNet variants with imbalance-aware loss functions performed well even without resampling, with PF-Loss achieving its second-highest G-mean (86.66) under the Original condition. Among

Table 20: G-mean by model and class imbalance strategy

Model	**Original**	**ROS**	**RUS**	**SMOTE**	**ROS & RUS**
Reg. LR	45.84	79.21	74.15	79.15	79.15
GB	49.88	84.93	**85.50**	83.63	83.69
XGB	45.79	85.30	85.43	83.46	83.26
AB	59.95	85.39	85.22	85.05	84.94
SVM (Lin.)	58.29	85.38	82.63	85.38	85.03
RF	63.96	85.27	84.56	84.86	84.63
CB	63.79	84.93	85.05	84.48	83.59
SVM (RBF)	58.31	83.21	84.65	82.64	82.67
DT	69.47	84.35	83.46	83.24	83.24
PF (ResNet)	**86.66**	85.64	84.50	**86.74**	86.50
WCE (ResNet)	85.57	**85.71**	85.20	85.87	85.51
FL (ResNet)	85.67	85.40	85.49	85.75	86.06
WLDAM (ResNet)	71.53	83.99	85.06	84.68	**86.41**

all model–strategy combinations, PF-Loss with SMOTE produced the highest single G-mean (86.74), while PF-Loss also performed best under the Original condition (86.66) and WLDAM performed best under ROS and RUS (86.41). For the traditional models, resampling strategies substantially improved performance, with oversampling yielding the highest average G-mean across all models.

In contrast to other traditional classifiers, whose G-mean drops sharply under the original no-resampling condition, three of the four tabular ResNet variants—WCE-Loss, Focal Loss, and PF-Loss—show minimal change in G-mean across the original and resampled settings. What this indicates is that cost-sensitive loss functions can internalize class imbalance handling during optimization. Consequently, external resampling appears largely redundant for these neural network models.

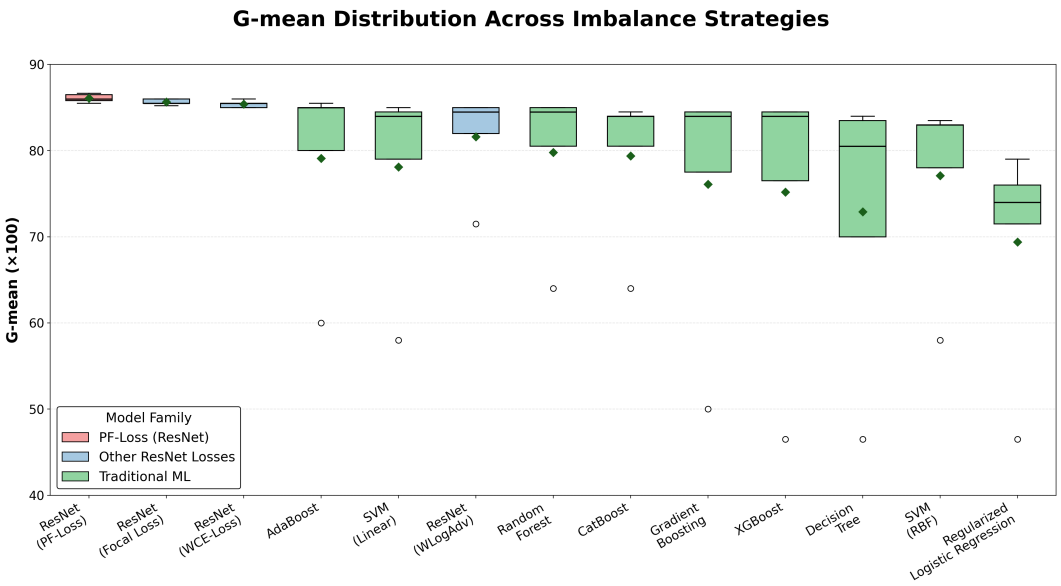


Figure 5: Distribution of G-mean scores (×100) across five class imbalance strategies for each classifier, ordered by median G-mean (descending).

5.5 Training Time Analysis

Table 21 presents a comparison of total training time, average time per configuration, number of evaluated configurations, and final G-mean across all models. The results emphasize the computational efficiency advantage of PF-Loss.

Table 21: Training time comparison across all models

Model	Total Time (s)	Avg per Config. (s)	# Config.	G-mean
Reg. LR	**2.7**	0.5	5	0.7786
RF	105.9	2.9	36	0.8423
XGB	172.2	3.2	54	0.8526
GB	2,711.1	50.2	54	0.8363
AB	231.5	5.8	40	0.8505
DT	19.0	**0.2**	90	0.8440
SVM (RBF)	8.7	1.0	9	0.6352
SVM (Lin.)	7.4	1.9	4	0.6039
CB	28.4	1.1	27	0.8487
PF (ResNet)	19.7	6.6	**3**	**0.8666**
WCE (ResNet)	73.1	6.1	12	0.8557
FL (ResNet)	189.1	7.0	27	0.8567
WLDAM (ResNet)	227.2	6.3	36	0.8505

PF-Loss required only 19.7 seconds of total training time, making it the fastest deep learning variant by a wide margin. Compared with Gradient Boosting (2,711.1 seconds), PF-Loss was approximately 138 times faster. Compared with WCE-Loss (73.1 seconds), the next-fastest neural network variant, PF-Loss was 3.7 times faster, and compared with WLDAM (227.2 seconds), the slowest, PF-Loss was 11.5 times faster. This computational advantage stemmed directly from its parameter-free formulation. PF-Loss required evaluation of only three architecture configurations, while WCE required 12, Focal Loss required 27, and WLDAM required 36. As a result, PF-Loss is well suited for settings where models must be deployed and retrained quickly as new data arrive, without the additional burden of loss-specific hyperparameter tuning.

The table reports the total number of unique hyperparameter combinations evaluated for each model during tuning. For the scikit-learn classifiers, these counts reflect the full Cartesian product of each model's search grid evaluated through GridSearchCV with stratified five-fold cross-validation. For example, Random Forest evaluated 36 configurations, Decision Tree evaluated 90, SVM evaluated 9 configurations for the RBF kernel and 4 for the linear kernel, and Regularized Logistic Regression evaluated five through the built-in regularization path in LogisticRegressionCV. CatBoost used a manual grid over depth, learning rate, and `l2_leaf_reg`, yielding 27 configurations with early stopping on a separate validation split. For the deep learning models, each loss function was paired with three candidate architectures; PF-Loss required only those three evaluations because it introduced no loss-specific hyperparameters, whereas WCE-Loss, Focal Loss, and WLDAM expanded the search to 12, 27, and 36 configurations, respectively.

5.6 SHAP Interpretability

SHAP (SHapley Additive exPlanations) analysis was used to identify the features most strongly associated with fraud predictions across both deep learning and tree-based models. For the neural network loss functions, SHAP values were computed for PF-Loss, which achieved the highest G-mean (0.8666), using `KernelExplainer` on a sample of 200 providers. For the tree-based models, `TreeExplainer` was applied to XGBoost and was used to identify consensus feature importance across six classifiers.

Table 22 reports the top consensus fraud indicators, defined as features that appeared most frequently among the top 15 importance rankings across the tree-based models. The most consistent indicators were provider-level and physician-level billing variables, especially average attending physician claim reimbursement and overall claim reimbursement, which appeared across all six models. Average provider claim reimbursement, average operating physician claim reimbursement, and average admit diagnosis claim reimbursement each appeared in five of the six models. Provider-level diagnosis code claim counts, admission duration, deductible amounts paid, and average attending physician admission days also appeared in four of the six models. This suggests that fraud predictions are driven primarily by billing behavior, particularly unusually high reimbursement amounts, elevated claim

volumes, and irregular financial patterns linked to physicians and diagnosis codes.

In contrast, chronic condition indicators and demographic variables such as age, gender, and race showed relatively low SHAP importance across models. This suggests that the models relied more on behavioral billing patterns than on patient characteristics. The consistency of these rankings supports audit prioritization based on anomalous reimbursement patterns, high claim volume, and suspicious provider-physician billing relationships.

Table 22: Top consensus fraud indicators across tree-based models

Feature	**Models (out of 6)**
Avg Attending Physician Claim Reimbursement	6/6
Claim Reimbursement ($)	6/6
Avg Provider Claim Reimbursement	5/6
Avg Operating Physician Claim Reimbursement	5/6
Avg Admit Diagnosis Claim Reimbursement	5/6
Provider-Diagnosis Claim Count	4/6
Admission Duration (Days)	4/6
Deductible Paid ($)	4/6
Avg Attending Physician AdmitForDays	4/6
Avg Operating Physician IP Deductible	3/6
Provider-Beneficiary Claim Count	3/6

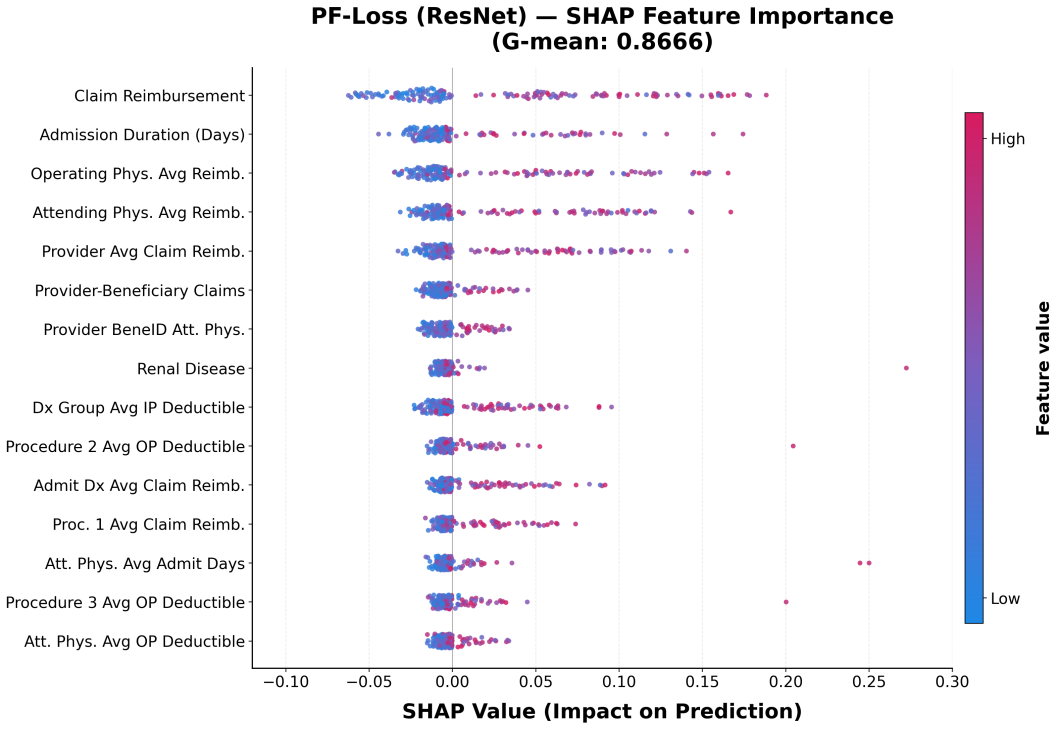


Figure 6: SHAP feature importance for PF-Loss

5.7 Quantitative Risk Analysis

Beyond binary classification, this thesis extends fraud detection outputs into a quantitative risk framework that converts predicted probabilities into dollar-denominated measures of risk. Drawing on Basel II/III credit risk methodology, the framework includes four components: probability calibration, expected loss estimation, budget-constrained audit optimization, and Monte Carlo simulation.

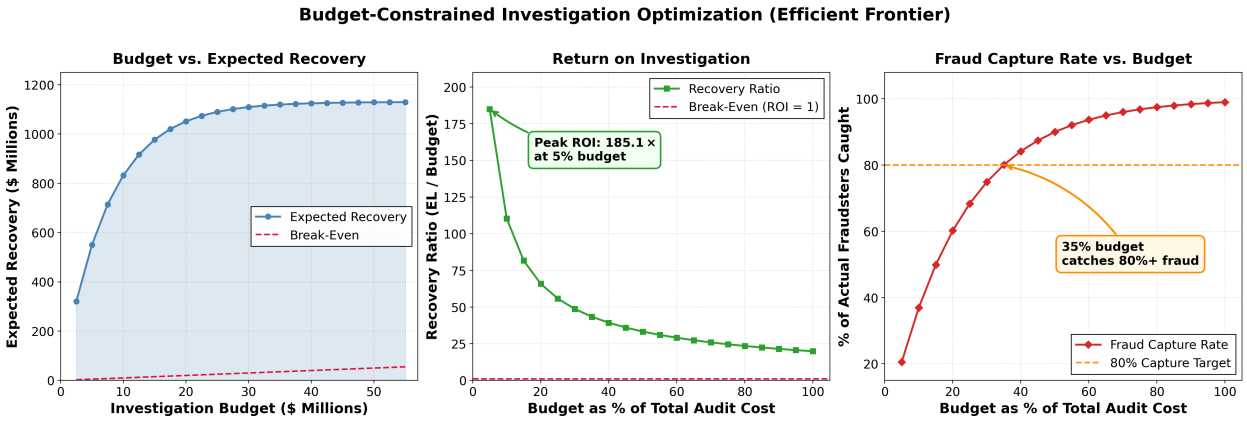


Figure 7: Budget-constrained investigation optimization: expected recovery, return on investment, and fraud capture rate across budget allocations.

Probability calibration was evaluated using the Brier Score, defined as the mean squared error between predicted probabilities and observed outcomes. In binary classification, a score of 0 indicates perfect calibration, whereas 0.25 reflects random predictions. Reliability diagrams were also used to compare predicted probabilities with observed fraud frequencies across bins. For downstream risk estimation, the best-calibrated model, identified by the lowest Brier Score, was selected because accurate probability estimates are more important than classification accuracy when outputs are converted into dollar-denominated risk measures.

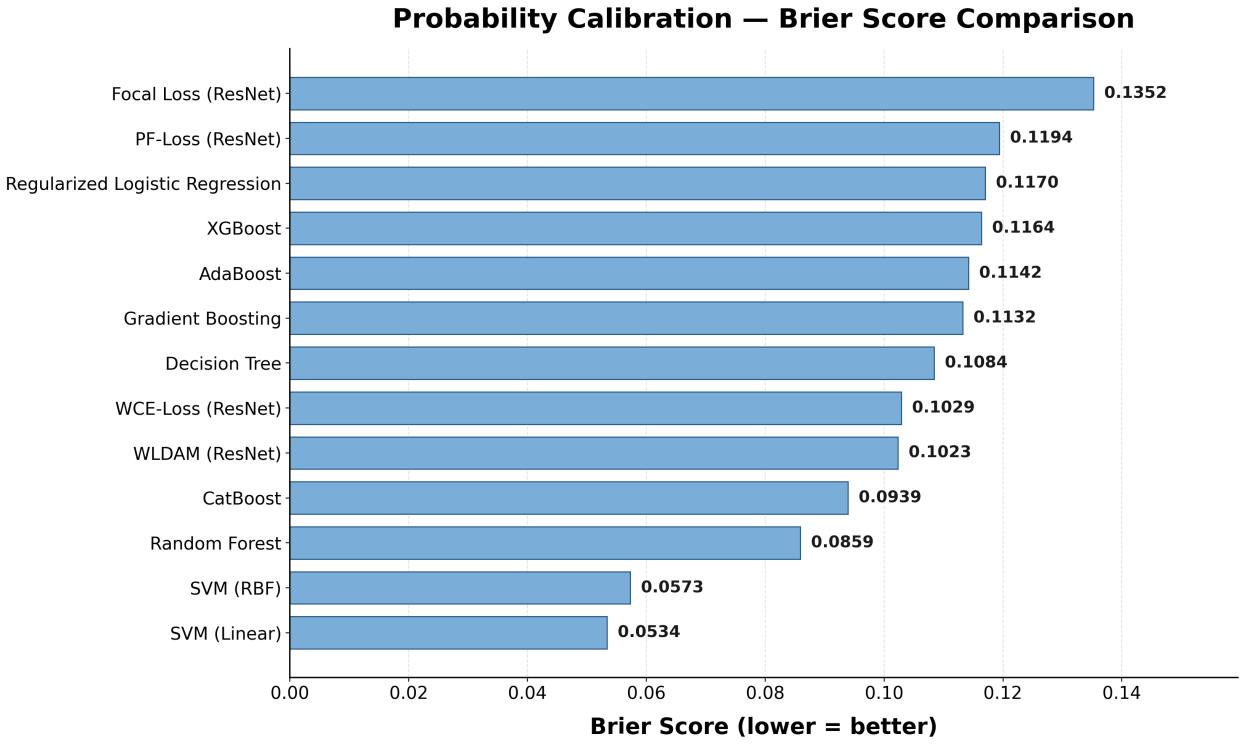


Figure 8: Brier Score comparison across all models

Following the Basel II/III credit risk methodology, the expected loss for each provider was calculated as

$$EL_i = P_i \times Exposure_i$$

where P_i is the model-predicted fraud probability and $Exposure_i$ is the total reimbursement amount. Risk-decile analysis revealed a highly concentrated loss distribution: the top 10% of providers by predicted risk accounted for the majority of the total expected loss, consistent with the Pareto principle. Across the full set of 5,410 providers, the total expected loss was estimated to be in the hundreds of millions of dollars, with the mean expected loss per provider differing sharply across deciles.

In practice, fraud investigation units face finite budgets and are unable to audit every flagged provider. To model this constraint, a 0-1 knapsack problem was formulated using linear programming through `scipy.optimize.linprog`, with the objective of maximizing expected loss captured under a fixed budget. Audit cost per provider was defined as a \$5,000 base amount plus \$50 per claim. The resulting efficient frontier demonstrated diminishing marginal returns on investment: initial budget allocations targeted the highest-risk providers and captured the greatest expected loss, while further increases in the budget yielded progressively smaller marginal gains.

Table 23: Quantitative risk measures and their applications

Risk Measure	**Description**	**Application**
Expected Loss (EL)	$P(\text{fraud}) \times \text{Exposure}$	Provider ranking by economic risk
VaR (95%)	95th percentile of loss distribution	Budget planning threshold
VaR (99%)	99th percentile of loss distribution	Worst-case scenario planning
CVaR (ES)	Mean loss beyond VaR threshold	Tail risk quantification
Gini Coefficient	EL concentration measure	Resource allocation efficiency
Efficient Frontier	Max EL captured per budget level	Optimal budget allocation

To quantify uncertainty in the expected loss estimates, 10,000 Monte Carlo simulations were conducted. For each simulation, provider fraud status was sampled from a Bernoulli distribution with probability P_i, and recovery rates were sampled from a Beta(8, 6) distribution with a mean of approximately 57%. The simulated loss distribution was then used to estimate Value-at-Risk (VaR) at the 95th and 99th percentiles, Expected Shortfall, also referred to as Conditional VaR, and confidence intervals for total system loss. Based on these results, providers were classified into Low, Moderate, High, and Critical risk tiers, with corresponding audit recommendations for each tier.

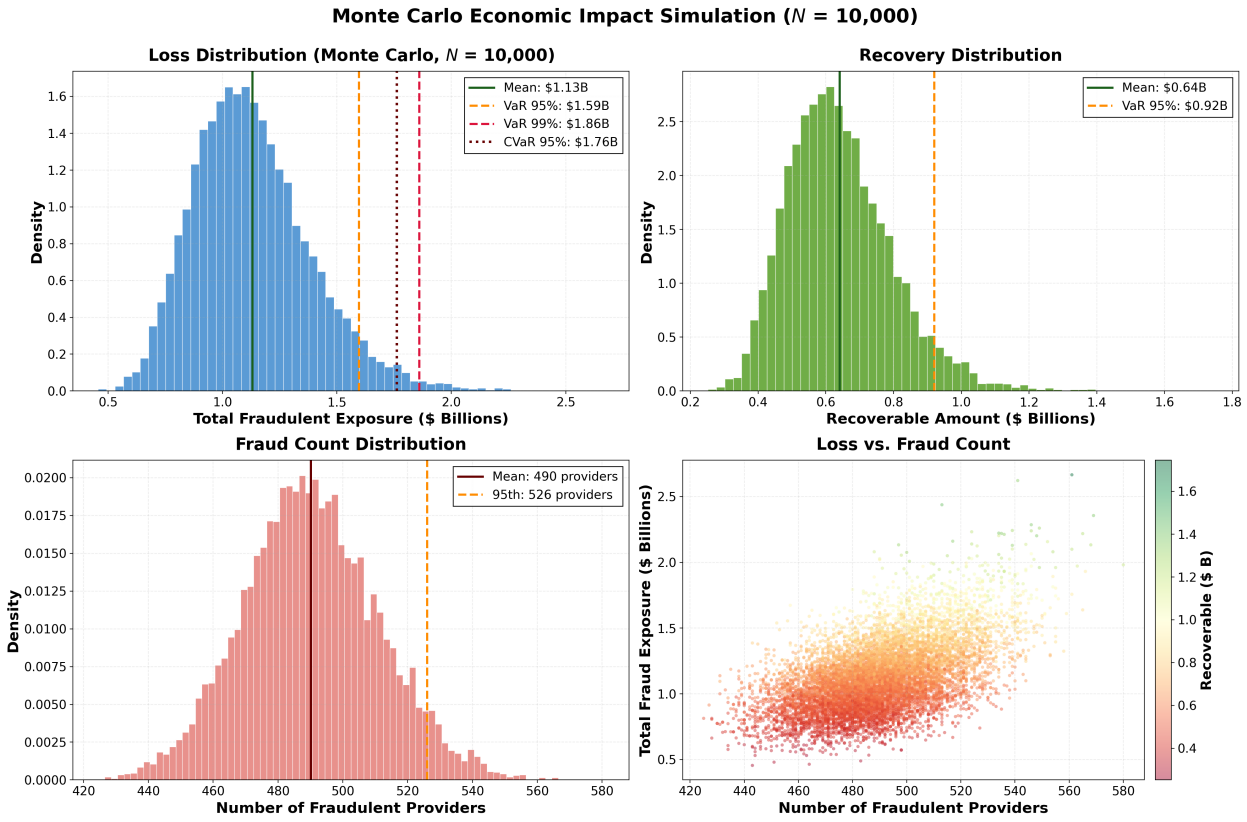


Figure 9: Monte Carlo economic impact simulation (N = 10,000): loss distribution with VaR/CVaR risk measures, recovery distribution, fraud count distribution, and loss-count joint scatter

5.8 Median-Based Expenditure Threshold Heuristic

To contextualize the value added by the supervised learning pipeline, this subsection evaluates a supplemental rule-based baseline derived directly from raw provider expenditure. The motivating question is whether a simple dollar-threshold rule could substitute for a trained classifier.

In this context, a *heuristic* refers to a fixed decision rule that does not involve machine learning. Unlike the supervised models in Sections 6.2–6.7, the heuristic does not learn from training data, fit any parameters, or use any of the 163 engineered features. Instead, it applies a single dollar-figure cutoff directly to each provider's total reimbursement: providers whose spending exceeds the cutoff are flagged as fraudulent, while all others are labeled non-fraudulent. This makes the rule fully transparent, immediately interpretable, and computationally free, but it also restricts the decision to one variable—raw expenditure—rather than the full feature set used by the supervised classifiers. Formally, the heuristic flags any provider whose total reimbursement exceeds the population median by a fixed offset Δ:

$$\hat{y}_i^{\text{heur}} = \mathbf{1}[\,\text{Exposure}_i > \text{median}(\text{Exposure}) + \Delta\,] \tag{18}$$

Setting Δ = \$20,000 (per the advisor's suggestion) yields a threshold of \$231,660 on top of the provider-level median exposure of \$211,660. Each provider's heuristic label $\hat{y}_i^{\text{heur}}$ was then compared to the ground-truth `PotentialFraud` indicator across all 5,410 providers, and the resulting sensitivity, specificity, and G-mean were benchmarked against the best-performing supervised model.

The heuristic captures 95.3% of fraudulent providers (sensitivity = 0.9526) but at the cost of flagging nearly half the population; only 18.8% of its alerts correspond to actual fraud (precision = 0.1881). The resulting G-mean of 0.7405 reflects the trade-off: strong recall offset by poor specificity, which is operationally untenable in a budget-constrained audit setting. Figure 10 sweeps the offset across $\Delta \in [-\$100\text{K}, +\$1\text{M}]$ and confirms that no fixed offset produces a balanced classifier; raising Δ improves precision but sacrifices sensitivity faster than specificity can compensate, with the heuristic's own G-mean peaking only modestly near Δ = +\$250K.

The best supervised model (PF-Loss, G-mean = 0.8666) improves on the heuristic by +0.1261 in absolute terms, or approximately +17.0% relative. As shown in Figure 10 by the dashed reference line, PF-Loss exceeds every threshold value evaluated in the sweep-not only the advisor-suggested Δ = +\$20K point, but also the heuristic's own optimum. This lift reflects the value added by combining 163 engineered billing features under a class-imbalance-aware loss objective, rather than relying on a single-dimensional thresholding rule. The heuristic remains useful as a transparent sanity check for downstream stakeholders—it confirms that high-spending providers are indeed disproportionately fraudulent—but the supervised pipeline meaningfully outperforms it on the central balance of detection and false-alarm avoidance.

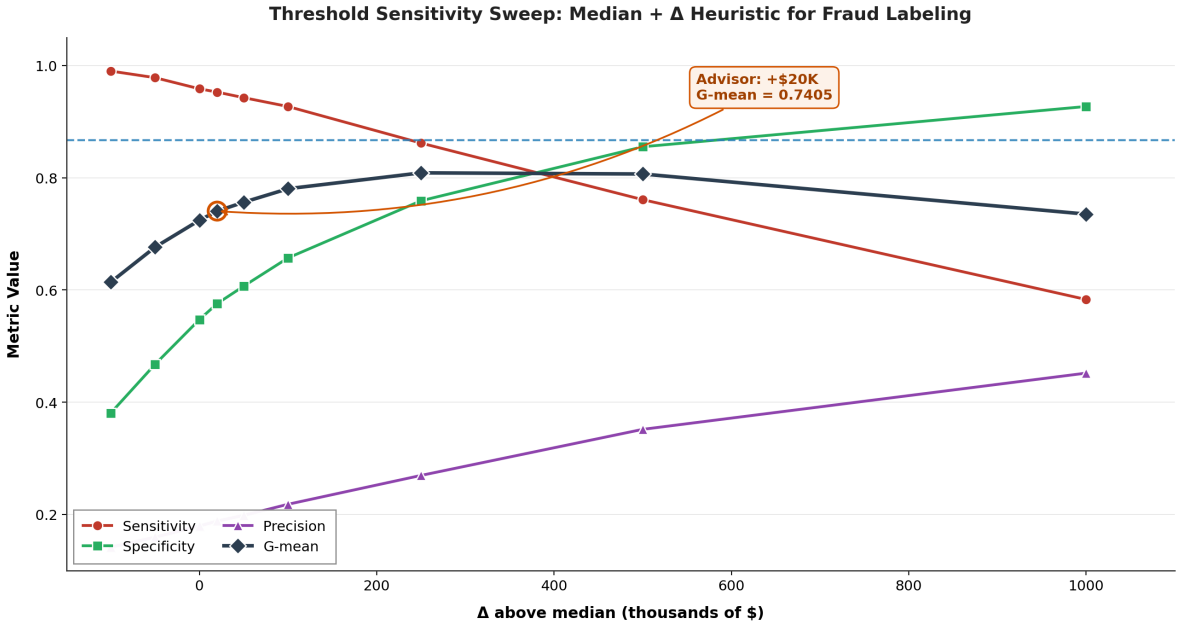


Figure 10: Threshold sensitivity sweep for the median-plus-Δ heuristic.

6 Discussion

6.1 Interpretation of Results

The central finding is that class-imbalance-aware loss functions, embedded within a tabular ResNet architecture, provide the strongest overall performance for Medicare provider fraud detection. Among the neural network variants, PF-Loss achieved the highest G-mean (0.8666), followed by Focal Loss (0.8567), WCE-Loss (0.8557), and WLDAM (0.8505). All four ResNet variants outperformed every traditional machine learning model, demonstrating the advantage of combining deep learning with imbalance-aware loss formulations for this task. PF-Loss is notable given its highest G-mean overall while requiring no additional hyperparameters for class-imbalance handling. PF-Loss trained in only 19.7 seconds, approximately 89% faster than XGBoost (172.2 seconds) and 81% faster than Random Forest (105.9 seconds). The class imbalance strategy analysis further highlights PF-Loss's inherent robustness: under the Original (no resampling) condition, traditional models such as XGBoost and Regularized Logistic Regression collapsed to G-mean values of 45.79 and 45.84, respectively, whereas PF-Loss maintained a G-mean of 86.66 without any external resampling. This reflects PF-Loss's recall for fraud cases without sacrificing specificity, a balance that is operationally important because missed fraudulent providers translate into ongoing financial losses.

Among the traditional models, XGBoost and CatBoost performed strongly, consistent with prior studies identifying gradient-boosted methods as reliable Medicare fraud classifiers [25, 13]. The additional benefit of combining PF-Loss with SMOTE was real but modest: the best SMOTE-augmented variant achieved a G-mean of 86.74, only 0.08 percentage points above PF-Loss alone (86.66), suggesting that PF-Loss already captures nearly all of the benefit typically associated with resampling. Notably, PF-Loss achieved its highest G-mean (86.66) under the Original condition without any resampling, suggesting that its class-average probability mechanism effectively addresses class imbalance through the loss function alone.

These results are consistent with prior work. The strong performance of tree-based ensembles aligns with Herland et al. [8], and the results support the conclusion of Leevy et al. [25] that supervised binary classification outperforms one-class approaches under severe imbalance. Johnson and Khoshgoftaar [14] found that data-level methods significantly outperformed algorithm-level methods at a 99.97:0.03 imbalance ratio; the present thesis finds that algorithm-level loss functions such as PF-Loss and Focal Loss perform competitively at a less severe 10:1 ratio, suggesting that the effectiveness of the strategy may depend on imbalance severity. Hancock et al. [13] demonstrated that data reduction techniques improve classification on imbalanced Medicare data; the present work explores the orthogonal strategy of modifying the loss function directly, and these complementary approaches could be combined in future work.

A key advantage of PF-Loss is its dynamic adaptation to class-level performance within each mini-batch. Unlike WCE, whose class weights are fixed before training, and Focal Loss,

whose modulating factor requires a manually tuned γ, PF-Loss responds directly to how well each class is being learned through the class-average probability mechanism. When average fraud-class predictions are poor, PF-Loss automatically increases gradient contributions for those samples; as performance improves, gradients diminish naturally. This adaptive behavior is beneficial given the 9.36% fraud prevalence, where random mini-batches contain few fraud samples. The result is a model that achieves balanced sensitivity (0.8684) and specificity (0.8647) without any loss-specific hyperparameter tuning—requiring only three architecture configurations compared with 12 for WCE, 27 for Focal Loss, and 36 for WLDAM.

A complementary baseline comparison reinforces the operational value of the supervised approach. A simple median+\$20K expenditure threshold flags any provider whose total reimbursement exceeds the population median by \$20,000 — a rule transparent enough to be implemented without any machine learning infrastructure. While this heuristic captures 95.3% of fraudulent providers, it does so by flagging 47.4% of the population, yielding a G-mean of 0.7405. The PF-Loss pipeline improves on this by +0.1261 in absolute terms (+17.0% relative), confirming that the lift from supervised modeling is meaningful and not merely a restatement of high-spender risk.

6.2 SHAP Interpretability Insights

The SHAP analysis indicates that fraud detection is driven primarily by billing behavior rather than patient demographics. Features related to average reimbursement per physician, provider-level reimbursement, admission duration, deductible amounts, and claim volume consistently ranked among the most important predictors across model families. Average attending physician claim reimbursement and overall claim reimbursement appeared in all six tree-based models, while average provider reimbursement, average operating physician reimbursement, and average admit diagnosis reimbursement each appeared in five of six. TreeSHAP provides exact Shapley values for tree-based models under the independence assumption [59], while SHAP analysis for the loss function variants used approximate `KernelExplainer` methods due to the #P-hardness of exact computation for neural networks.

High average reimbursement per claim, particularly at the attending and operating physician level, confirms billing inflation as a primary fraud indicator. Provider-diagnosis claim counts and provider-beneficiary claim counts suggest that providers with concentrated billing patterns across narrow diagnosis codes or specific beneficiaries may reflect upcoding or phantom billing arrangements. Admission duration appeared in four of six models, consistent with the expectation that inflated or unnecessary inpatient stays generate disproportionate reimbursement. Demographic variables and chronic condition indicators had low importance, suggesting the models capture financial irregularities rather than demographic proxies. These findings imply that effective fraud screening may be possible with a compact set of billing-pattern variables. In operational CMS settings processing millions of records, TreeSHAP provides a computationally feasible path to investigator support and audit triage.

6.3 Limitations

It's important to note a series of limitations: First, the Kaggle fraud labels, derived from CMS exclusion records, likely capture only a subset of true fraud cases, introducing label noise. Hancock et al. [13] note that the LEIE does not include reinstated providers, and the labels do not distinguish among fraud types. Second, the 9.36% fraud rate is substantially higher than the 0.03–0.07% rates in administrative CMS datasets [14, 13], so results may not transfer directly to more severely imbalanced settings. Third, the dataset represents a historical snapshot; Johnson and Khoshgoftaar [4] documented new fraud forms emerging during COVID-19, and the pipeline does not explicitly account for concept drift. Fourth, provider-level aggregation may obscure claim-level or temporal patterns that are also informative; Matloob et al. [51] showed that transaction-level sequence analysis can detect anomalies invisible at the aggregate level. Fifth, the standard train/tune/validation splits do not fully address provider-memorization concerns raised by Johnson and Khoshgoftaar [4], who recommended k-fold-by-NPI cross-validation. Sixth, the quantitative risk framework relies on simplifying assumptions about audit costs and recovery rates, requiring calibration against operational data before deployment.

6.4 Implications and Contributions

This thesis demonstrates that PF-Loss, previously validated only on image-classification benchmarks, achieves the strongest performance on tabular healthcare fraud data without any hyperparameter tuning for class imbalance. Viewed through the framework of Jordan and Mitchell [33], PF-Loss represents the kind of new formalization that advances the field: it reformulates cost-sensitive learning at the class level, eliminating manually tuned imbalance parameters while achieving the highest G-mean among all models. PF-Loss achieved a G-mean of 0.8666, surpassing XGBoost (0.8526) while training in a fraction of the time. The broader comparison across all four loss functions reflects what Jordan and Mitchell identify as central to practical deployment decisions: Focal Loss achieved the second-highest G-mean (0.8567) but required 27 configurations and 189.1 seconds of training, whereas PF-Loss required only 3 configurations and 19.7 seconds. Their discussion of privacy and fairness is also relevant: providers flagged for investigation have both legal and ethical interests in understanding the basis for that determination, and the SHAP-based explanations provide a human-interpretable account of model decisions in terms of specific billing behaviors.

The study further contributes a SHAP-based interpretability analysis, identifying actionable fraud indicators, and a Monte Carlo risk simulation, translating classification outputs into dollar-denominated financial measures for audit decision support. Overall, the results suggest that parameter-free imbalance learning is a promising direction for Medicare fraud detection. A tabular ResNet paired with PF-Loss can outperform heavily tuned gradient-boosted ensembles with much lower tuning overhead than conventional imbalance-aware losses, representing a practical and credible operational approach for agencies that need interpretable, efficient, and repeatable fraud screening tools.

7 Conclusion

7.1 Summary of Findings

This thesis presents among the first applications of the Parameter-Free Loss (PF-Loss) function [21] to Medicare fraud detection. Using publicly available CMS claims data aggregated to 5,410 providers, 558,211 claims, and 163 engineered features, the study evaluated twelve supervised models, eight traditional and ensemble classifiers and four tabular ResNet deep learning variants, under a consistent framework with G-mean as the primary metric.

Among the neural network loss variants, PF-Loss obtained the highest G-mean (0.8666), followed by Focal Loss (0.8567), WCE-Loss (0.8557), and WLDAM (0.8505). All four ResNet variants outperformed every traditional machine learning model in G-mean. PF-Loss outperformed all traditional models, including the best traditional model, XGBoost (0.8526), while requiring no loss-specific hyperparameters and only 19.7 seconds of training time, approximately 138 times faster than Gradient Boosting, 11.5 times faster than WLDAM, and 3.7 times faster than WCE-Loss. A class imbalance strategy comparison showed that traditional models collapsed to G-mean values as low as 45.79 under the Original (no resampling) condition, whereas PF-Loss maintained a G-mean of 86.66 without any external resampling. Combining PF-Loss with SMOTE yielded a modest improvement to 86.74, only 0.08 percentage points above PF-Loss alone, suggesting that PF-Loss already captures nearly all of the benefit typically associated with resampling. Among the traditional models, XGBoost and CatBoost performed the strongest, consistent with Leevy et al. [25] and Hancock et al. [13].

The SHAP-based interpretability analysis consistently identified reimbursement amounts at the provider and physician levels, deductible amounts, claim counts, admission duration, and more as aligning closely with known Medicare fraud patterns. The quantitative risk framework extended predictions into expected loss estimation, budget-constrained audit prioritization, and Monte Carlo uncertainty analysis, translating classification performance into dollar-denominated decision support.

7.2 Key Contributions and Practical Implications

This thesis makes five contributions: **(1)** Demonstrating the viability of PF-Loss for tabular healthcare fraud detection beyond its original image-classification setting, **(2)** providing a unified 12-model benchmarking framework under consistent preprocessing and evaluation, **(3)** developing a reproducible, leakage-aware preprocessing pipeline, **(4)** contributing SHAP-based interpretability that identifies actionable fraud indicators, **(5)** introducing a quantitative risk framework that converts fraud probabilities into expected losses and provides audit decision support.

PF-Loss serves as a stand-in for standard cross-entropy without the tuning overhead required by WCE, Focal Loss, or LDAM, making it a strong default when training speed, reproducibility, and limited tuning resources are priorities. While Hancock et al. [13] showed that data reduction techniques improve classification on imbalanced Medicare data, the present work demonstrates that algorithm-level intervention through imbalance-aware loss functions provides complementary gains without modifying the training distribution. The SHAP analysis suggests that fraud screening may be achievable with a compact subset of billing-pattern variables. The efficient frontier analysis shows that targeting the highest-risk providers first captures the greatest expected loss under constrained audit budgets.

7.3 Future Work

Several directions follow from this study. **Temporal validation:** Testing PF-Loss on consecutive CMS data releases would assess stability under concept drift, particularly given the new fraud forms documented by Johnson and Khoshgoftaar [4] during COVID-19. **Extreme imbalance:** Evaluation on datasets with 1,000:1 or greater imbalance ratios [14] would test PF-Loss's gradient mechanism under conditions where mini-batches may contain no minority-class samples. **Multiclass modeling:** Extending PF-Loss to distinguish fraud subtypes could support more targeted audit strategies. **Advanced architectures:** Graph neural networks for provider-beneficiary relationships, temporal models for longitudinal claims, and transaction-level sequence mining [51] could complement provider-level classification. **Feature selection:** The interaction between ensemble feature selection [13] and PF-Loss's class-level gradients remains an open question. **Validation and deployment:** Provider-aware k-fold-by-NPI cross-validation [4], fairness evaluation across specialties and geographies, and integration with real-time claims processing and SHAP-based audit dashboards would support the transition from retrospective modeling to operational fraud detection.

7.4 Final Remarks

PF-Loss produced the highest G-mean in this study, surpassing both the best traditional model (XGBoost) and all other neural network variants, while substantially reducing the training and tuning burden, requiring only three architecture configurations and 19.7 seconds compared with 27 configurations and 189.1 seconds for the next-best Focal Loss. This efficiency advantage matters in applied fraud detection, where models must be retrained, interpreted, and deployed under operational constraints. By integrating predictive modeling, interpretability analysis, and risk-based decision support into a single framework, this thesis offers an approach that is both methodologically significant and practically applicable to healthcare fraud detection.

Bibliography

[1] National Health Care Anti-Fraud Association. *The Challenge of Health Care Fraud*. https://www.nhcaa.org/tools-insights/about-health-care-fraud/the-challenge-of-health-care-fraud/. Accessed: 2025. 2023.

[2] Centers for Medicare and Medicaid Services. *Medicare Fraud and Abuse: Prevention, Detection, and Reporting*. https://www.cms.gov/. Accessed: 2025. 2019.

[3] Centers for Medicare and Medicaid Services. *Medicare Provider Utilization and Payment Data*. https://www.cms.gov/. Accessed: 2025. 2024.

[4] Justin M. Johnson and Taghi M. Khoshgoftaar. "Data-Centric AI for Healthcare Fraud Detection". In: *SN Computer Science* 4.389 (2023), pp. 1–23. DOI: 10.1007/s42979-023-01809-x.

[5] U.S. Department of Justice. *National Health Care Fraud Enforcement Action*. https://www.justice.gov/. Accessed: 2025. 2023.

[6] OKPublicData. *Oklahoma Public Data Resources*. https://okpublicdata.com/. Accessed: 2025. 2024.

[7] Richard A. Bauder and Taghi M. Khoshgoftaar. "The Detection of Medicare Fraud Using Machine Learning Methods with Excluded Provider Labels". In: *Proceedings of the 31st International Florida Artificial Intelligence Research Society Conference (FLAIRS)*. AAAI Press, 2018, pp. 404–409.

[8] Matthew Herland, Taghi M. Khoshgoftaar, and Richard A. Bauder. "Big Data Fraud Detection Using Multiple Medicare Data Sources". In: *Journal of Big Data* 5.29 (2018), pp. 1–21. DOI: 10.1186/s40537-018-0138-3.

[9] Hossein Joudaki et al. "Using Data Mining to Detect Health Care Fraud and Abuse: A Review of Literature". In: *Global Journal of Health Science* 7.1 (2015), pp. 194–202. DOI: 10.5539/gjhs.v7n1p194.

[10] Haibo He and Edwardo A. Garcia. "Learning from Imbalanced Data". In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), pp. 1263–1284. DOI: 10.1109/TKDE.2008.239.

[11] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. "Fraud Detection System: A Survey". In: *Journal of Network and Computer Applications* 68 (2016), pp. 90–113. DOI: 10.1016/j.jnca.2016.04.007.

[12] Jarrod West and Maumita Bhatt. "Intelligent Financial Fraud Detection: A Comprehensive Review". In: *Computers & Security* 57 (2016), pp. 47–66. DOI: 10.1016/j.cose.2015.09.005.

[13] John T. Hancock et al. "Data Reduction Techniques for Highly Imbalanced Medicare Big Data". In: *Journal of Big Data* 11.8 (2024), pp. 1–30. DOI: 10.1186/s40537-023-00869-3.

[14] Justin M. Johnson and Taghi M. Khoshgoftaar. "Medicare Fraud Detection Using Neural Networks". In: *Journal of Big Data* 6.63 (2019), pp. 1–35. DOI: 10.1186/s40537-019-0225-0.

[15] Nitesh V. Chawla et al. "SMOTE: Synthetic Minority Over-Sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.

[16] Yin Cui et al. "Class-Balanced Loss Based on Effective Number of Samples". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9260–9269. DOI: 10.1109/CVPR.2019.00949.

[17] Joanito Agili Lopo and Kristoko Dwi Hartomo. "Evaluating Sampling Techniques for Healthcare Insurance Fraud Detection in Imbalanced Dataset". In: *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika* 9.2 (2023), pp. 223–238.

[18] Chen Huang et al. "Learning Deep Representation for Imbalanced Classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5375–5384. DOI: 10.1109/CVPR.2016.580.

[19] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (2020), pp. 318–327. DOI: 10.1109/TPAMI.2018.2858826.

[20] Kaidi Cao et al. "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2019.

[21] Jie Du et al. "Parameter-Free Loss for Class-Imbalanced Deep Learning in Image Classification". In: *IEEE Transactions on Neural Networks and Learning Systems* 34.6 (2023), pp. 3234–3240. DOI: 10.1109/TNNLS.2021.3110885.

[22] Tom Fawcett and Foster Provost. "Adaptive Fraud Detection". In: *Data Mining and Knowledge Discovery* 1.3 (1997), pp. 291–316. DOI: 10.1023/A:1009700419189.

[23] Richard J. Bolton and David J. Hand. "Statistical Fraud Detection: A Review". In: *Statistical Science* 17.3 (2002), pp. 235–255. DOI: 10.1214/ss/1042727940.

[24] Peter Travaille et al. "Electronic Fraud Detection in the U.S. Medicaid Healthcare Program: Lessons Learned from Other Industries". In: *Proceedings of the 17th Americas Conference on Information Systems (AMCIS)*. 2011, pp. 1–11.

[25] Joffrey L. Leevy et al. "Investigating the Effectiveness of One-Class and Binary Classification for Fraud Detection". In: *Journal of Big Data* 10.157 (2023), pp. 1–17. DOI: 10.1186/s40537-023-00825-1.

[26] John T. Hancock III et al. "A New and Effective Technique for Unsupervised Labeling and Feature Selection with Applications in Healthcare Fraud Detection". In: *2025 IEEE International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2025, pp. 301–308. DOI: 10.1109/IRI66576.2025.00063.

[27] Anli du Preez et al. "Fraud Detection in Healthcare Claims Using Machine Learning: A Systematic Review". In: *Artificial Intelligence in Medicine* 160 (2025), p. 103061. DOI: 10.1016/j.artmed.2024.103061.

[28] Ethan D. Curtis et al. "A Review of Distinct Machine Learning Classifiers for Healthcare Fraud Detection". In: *Journal of Big Data* 12.238 (2025), pp. 1–38. DOI: 10.1186/s40537-025-01295-3.

[29] Natalie Thorpe et al. "Combating Medicare Fraud: A Struggling Work in Progress". In: *Franklin Business & Law Journal* 2012.4 (2012), pp. 95–107.

[30] Alberto Coustasse et al. "Upcoding Medicare: Is Healthcare Fraud and Abuse Increasing?" In: *Perspectives in Health Information Management* 18.4 (2021), 1f.

[31] Vivek Pande and Will Maas. "Physician Medicare Fraud: Characteristics and Consequences". In: *International Journal of Pharmaceutical and Healthcare Marketing* 7.1 (2013), pp. 8–33. DOI: 10.1108/17506121311315391.

[32] Chelsea Hill et al. "Medicare Fraud in the United States: Can It Ever Be Stopped?" In: *The Health Care Manager* 33.3 (2014), pp. 254–260. DOI: 10.1097/HCM.0000000000000019.

[33] Michael I. Jordan and Tom M. Mitchell. "Machine Learning: Trends, Perspectives, and Prospects". In: *Science* 349.6245 (2015), pp. 255–260. DOI: 10.1126/science.aaa8415.

[34] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[35] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[36] Liudmila Prokhorenkova et al. "CatBoost: Unbiased Boosting with Categorical Features". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 31. 2018.

[37] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.

[38] Yoav Freund and Robert E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139. DOI: 10.1006/jcss.1997.1504.

[39] John T. Hancock et al. "Explainable Machine Learning Models for Medicare Fraud Detection". In: *Journal of Big Data* 10.154 (2023), pp. 1–23. DOI: 10.1186/s40537-023-00821-5.

[40] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". In: *Machine Learning* 20.3 (1995), pp. 273–297. DOI: 10.1007/BF00994018.

[41] Jinglong Li et al. “A Survey on Statistical Methods for Health Care Fraud Detection”. In: *Health Care Management Science* 11.3 (2008), pp. 275–287. DOI: 10.1007/s10729-007-9045-4.

[42] Rachel Bennett, Stephanie L. Pierce, and Talayeh Razzaghi. “Interpretable Machine Learning Models for Predicting Cesarean Delivery in Class III Obese Cohorts”. In: *IEEE Access* 12 (2024), pp. 1–15. DOI: 10.1109/ACCESS.2024.0429000.

[43] Bharat Richhariya and Mohammad Tanveer. “A Comprehensive Survey on Support Vector Machine Applications in Medical Diagnosis”. In: *SN Computer Science* 1.283 (2020), pp. 1–22. DOI: 10.1007/s42979-020-00327-0.

[44] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep Learning”. In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[45] Jinwon An and Sungzoon Cho. “Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability”. In: *Special Lecture on IE* 2.1 (2015), pp. 1–18.

[46] Mansour Zoubeirou A. Mayaki and Michel Riveill. “Multiple Inputs Neural Networks for Fraud Detection”. In: *Proceedings of the 2022 International Conference on Machine Learning, Control, and Robotics (MLCR)*. Suzhou, China, 2022.

[47] James Bergstra and Yoshua Bengio. “Random Search for Hyper-Parameter Optimization”. In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305.

[48] Jia Wu et al. “Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization”. In: *Journal of Electronic Science and Technology* 17.1 (2019), pp. 26–40. DOI: 10.11989/JEST.1674-862X.80904120.

[49] Lisha Li et al. “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization”. In: *Journal of Machine Learning Research* 18.185 (2018), pp. 1–52.

[50] Stefan Falkner, Aaron Klein, and Frank Hutter. “BOHB: Robust and Efficient Hyperparameter Optimization at Scale”. In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Vol. 80. PMLR. PMLR, 2018.

[51] Irum Matloob et al. “A Sequence Mining-Based Novel Architecture for Detecting Fraudulent Transactions in Healthcare Systems”. In: *IEEE Access* 10 (2022), pp. 48447–48463. DOI: 10.1109/ACCESS.2022.3170888.

[52] Li Li et al. “Parameter-Free Extreme Learning Machine for Imbalanced Classification”. In: *Neural Processing Letters* 52.3 (2020), pp. 1927–1944. DOI: 10.1007/s11063-020-10282-z.

[53] Qi Qian et al. “DR Loss: Improving Object Detection by Distributional Ranking”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12163–12172. DOI: 10.1109/CVPR42600.2020.01218.

[54] Kean Chen et al. “Towards Accurate One-Stage Object Detection with AP-Loss”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5119–5127. DOI: 10.1109/CVPR.2019.00526.

[55] Liwen Xu, Huaguang Zhu, and Jiali Chen. “Imbalanced Graph Learning via Mixed Entropy Minimization”. In: *Scientific Reports* 14.24892 (2024), pp. 1–13. DOI: 10.1038/s41598-024-75999-6.

[56] Xinyu Guo et al. “Automated Loss Function Search for Class-Imbalanced Node Classification”. In: *Proceedings of the 41st International Conference on Machine Learning (ICML)*. Vol. 235. PMLR. PMLR, 2024.

[57] Mingchen Li et al. “AutoBalance: Optimized Loss Functions for Imbalanced Data”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2021.

[58] Scott M. Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.

[59] Guy Van den Broeck et al. “On the Tractability of SHAP Explanations”. In: *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2021, pp. 6505–6513.

[60] Rohit Anand Gupta. *Healthcare Provider Fraud Detection Analysis*. `https://www.kaggle.com/datasets/rohitrox/healthcare-provider-fraud-detection-analysis`. Accessed: 2025. 2020.

Appendices

Supplementary Dataset Statistics

Table 24: Variable descriptions by domain

Variable	Type	Description
Target & Identifiers		
PotentialFraud	Binary	Provider-level fraud label (Yes/No)
Provider	ID	Unique provider identifier
BeneID, ClaimID	ID	Beneficiary and claim identifiers
Beneficiary Demographics		
Gender	Binary	1 = Male, 2 = Female
Race	Categorical	Coded 1–5 (1 = White, 84.5% of beneficiaries)
DOB, DOD	Date	Date of birth; date of death (missing if alive)
Age	Continuous	Derived from DOB; mean = 73.6, SD = 12.7, range [26, 101]
WhetherDead	Binary	Derived: 1 if DOD present, 0 otherwise
State, County	Categorical	Beneficiary geographic location
Chronic Condition Indicators (11 binary flags, recoded: 1 = Yes, 0 = No)		
ChronicCond_Diabetes	Binary	60.2% prevalence
ChronicCond_IschemicHeart	Binary	67.6% prevalence
ChronicCond_Heartfailure	Binary	49.4% prevalence
ChronicCond_Depression	Binary	35.6% prevalence
ChronicCond_Alzheimer	Binary	33.2% prevalence
ChronicCond_KidneyDisease	Binary	31.2% prevalence
ChronicCond_Osteoporasis	Binary	27.5% prevalence
ChronicCond_rheumatoidarthritis	Binary	25.7% prevalence
ChronicCond_ObstrPulmonary	Binary	23.7% prevalence
RenalDiseaseIndicator	Binary	14.1% prevalence
ChronicCond_Cancer	Binary	12.0% prevalence
ChronicCond_stroke	Binary	7.9% prevalence
Medicare Coverage & Financial		
NoOfMonths_PartACov	Continuous	Months of Part A coverage (0–12)
NoOfMonths_PartBCov	Continuous	Months of Part B coverage (0–12)

Continued on next page

Table 24 (continued)

Variable	Type	Description
IPAnnualReimbursementAmt	Continuous	Inpatient annual reimb.; mean = $3,660 (SD = $9,569)
IPAnnualDeductibleAmt	Continuous	Inpatient annual deductible; mean = $400 (SD = $956)
OPAnnualReimbursementAmt	Continuous	Outpatient annual reimb.; mean = $1,298 (SD = $2,494)
OPAnnualDeductibleAmt	Continuous	Outpatient annual deductible
Claim-Level Fields (Inpatient & Outpatient)		
InscClaimAmtReimbursed	Continuous	Per-claim insurance reimbursement
DeductibleAmtPaid	Continuous	Per-claim deductible; inpatient only
AttendingPhysician	Categorical	Attending physician ID (0.3% missing)
OperatingPhysician	Categorical	Operating physician ID (79.5% missing; structural)
OtherPhysician	Categorical	Other physician ID (64.2% missing; structural)
ClmDiagnosisCode_1--10	Categorical	ICD diagnosis codes; slots 1–4 most populated
ClmProcedureCode_1--6	Categorical	ICD procedure codes; 4.2% usage (slot 1) to 0% (slot 6)
ClmAdmitDiagnosisCode	Categorical	Admission diagnosis; inpatient only
DiagnosisGroupCode	Categorical	DRG code; inpatient only
AdmissionDt, DischargeDt	Date	Inpatient only; used to derive AdmitForDays
AdmitForDays	Continuous	Derived length of stay; range [1, 36]

Table 25: Beneficiary-level descriptive statistics (Train, N = 138,556)

Variable	**Level / Statistic**	**N or Value**	**%**
Demographics			
Race	White (1)	117,057	84.5%
	Black (2)	13,538	9.8%
	Hispanic (3)	5,059	3.7%
	Other (5)	2,902	2.1%
Age (Years)	Mean (SD)	73.6 (12.7)	Range [26, 101]
Age Group	<65	21,220	15.3%
	65–74	48,980	35.4%
	75–84	43,253	31.2%
	85+	25,103	18.1%
Financial			
IP Annual Reimb. ($)	Mean (SD)	$3,660 ($9,569)	Range [–$8,000, $161,470]
IP Annual Reimb. Bins	$0 or less	102,526	74.0%
	$1–$2k	1,186	0.9%
	$2k–$10k	18,027	13.0%
	>$10k	16,817	12.1%
IP Annual Deductible ($)	Mean (SD)	$400 ($956)	Range [$0, $38,272]
OP Annual Reimb. ($)	Mean (SD)	$1,298 ($2,494)	—
Chronic Conditions (prevalence, descending)			
Ischemic Heart	Yes	93,644	67.6%
Diabetes	Yes	83,391	60.2%
Heart Failure	Yes	68,402	49.4%
Depression	Yes	49,260	35.6%
Alzheimer	Yes	46,026	33.2%
Kidney Disease	Yes	43,279	31.2%
Osteoporosis	Yes	38,059	27.5%
Rheumatoid Arthritis	Yes	35,584	25.7%
Obstr. Pulmonary	Yes	32,859	23.7%
Renal Disease	Yes	19,578	14.1%
Cancer	Yes	16,621	12.0%
Stroke	Yes	10,954	7.9%

Table 26: Dataset dimensions after each merging and engineering stage

Stage	**Records**	**Variables**	**Description**
Inpatient ∪ Outpatient	558,211	31	Outer join on 28 shared claim columns
+ Beneficiary merge	558,211	57	Inner join on `BeneID`
+ Provider labels	558,211	58	Inner join on `Provider`
+ Feature engineering	558,211	165+	Group-mean and claim-count features
Provider-level aggregation	**5,410**	**163**	`groupby(Provider).sum()`; final modeling unit

PF-Loss Implementation Pseudocode

The following pseudocode describes the implementation of the Parameter-Free Loss (PF-Loss) function proposed by Du et. al. (2023) and its integration into the tabular ResNet training pipeline used in this thesis. PF-Loss eliminates all class-imbalance hyperparameters by normalizing each class's contribution to the loss by its per-batch sample count, ensuring that every class exerts equal influence on gradient updates regardless of frequency.

Algorithm 1 PF-Loss Computation

Input: $\mathbf{Z} \in \mathbb{R}^{B \times M}$: raw logits for B samples across M classes
Input: $\mathbf{y} \in \{0, \ldots, M-1\}^B$: ground-truth class labels
Input: $\epsilon = 10^{-8}$: numerical stability constant
Output: Scalar loss value $\mathcal{L}_{\text{PF}}$

$\mathbf{P} \leftarrow \textsc{Softmax}(\mathbf{Z}, \text{dim} = 1)$ ▷ convert logits to class probabilities
$M \leftarrow$ number of columns in $\mathbf{Z}$ ▷ total class count (M=2 for binary)
$\mathcal{L} \leftarrow 0$
for $j = 0$ **to** $M - 1$ **do**
  mask $\leftarrow (\mathbf{y} = j)$ ▷ boolean mask for samples in class j
  $N_j \leftarrow$ number of $\textsc{True}$ values in mask
  if $N_j > 0$ **then**
    $q_j \leftarrow \textsc{Mean}\big(\mathbf{P}[\text{mask}, j]\big)$ ▷ Eq. 10: $q_j = \frac{1}{N_j} \sum_{i \in C_j} p_{i,j}$
    $\mathcal{L} \leftarrow \mathcal{L} + \log(q_j + \epsilon)$
  else
    continue ▷ class absent: q_j=1, $\log(1)$=0
  end if
end for
return $-\mathcal{L}/M$ ▷ Eq. 9: $\mathcal{L}_{\text{PF}} = -\frac{1}{M} \sum_{j=1}^{M} \log(q_j)$

Note. The denominator is always M (total class count), not the number of classes present in the mini-batch. When class j is absent from a batch, its contribution is zero, but the normalization constant remains unchanged. This design requires *no hyperparameters*—no class weights, no focusing parameter, and no margin constants.

Algorithm 2 ResNet Training Pipeline with PF-Loss

Input: $\mathbf{X}_{\text{train}}$, $\mathbf{y}_{\text{train}}$: scaled training features and labels
Input: $\mathbf{X}_{\text{val}}$, $\mathbf{y}_{\text{val}}$: scaled validation features and labels
Input: $\theta = \{d_{\text{block}}, d_{\text{hidden_mult}}, n_{\text{blocks}}, \text{dropout}, \text{lr}, \text{batch_size}\}$
Output: Trained ResNet model with best validation G-mean

/ Initialization */*
Set random seeds for reproducibility
model $\leftarrow$ BuildTabularResNet(θ) ▷
optimizer $\leftarrow$ Adam(model.params, lr=θ.lr, wd=10^{-4})
scheduler $\leftarrow$ ReduceLROnPlateau(patience=5, factor=0.5)
loss_fn $\leftarrow$ PFLoss() ▷ no hyperparameters to configure
$G^* \leftarrow -\infty$; $c \leftarrow 0$; $P \leftarrow 10$ ▷ best G-mean, patience counter, limit

/ Training loop */*
for epoch = 1 **to** max_epochs **do**
  model.train()
  for each mini-batch $(\mathbf{X}_b, \mathbf{y}_b)$ **in** DataLoader($\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}$) **do**
    $\hat{\mathbf{Z}} \leftarrow \text{model}(\mathbf{X}_b)$ ▷ forward pass: raw logits
    $\mathcal{L} \leftarrow \text{ComputePFLoss}(\hat{\mathbf{Z}}, \mathbf{y}_b)$ ▷ Algorithm 1
    Backpropagate $\mathcal{L}$; optimizer.step()
  end for

  / Validation and early stopping */*
  model.eval()
  $\mathbf{p} \leftarrow \text{Softmax}\big(\text{model}(\mathbf{X}_{\text{val}}), \text{dim=1}\big)[:, 1]$
  $\hat{\mathbf{y}} \leftarrow (\mathbf{p} > 0.5)$
  $G \leftarrow \sqrt{\text{Sensitivity}(\mathbf{y}_{\text{val}}, \hat{\mathbf{y}}) \times \text{Specificity}(\mathbf{y}_{\text{val}}, \hat{\mathbf{y}})}$
  scheduler.step(G)

  if $G > G^*$ **then**
    $G^* \leftarrow G$; save model state; $c \leftarrow 0$
  else
    $c \leftarrow c + 1$
    if $c \geq P$ **then**
      break
    end if
  end if
end for

Restore model to best saved state
return model, G^*

Note: The architecture search evaluates multiple ResNet configurations (varying d_{block}, n_{blocks}, $d_{\text{hidden_multiplier}}$, dropout, learning rate, and batch size) with PF-Loss as the sole loss function. PF-Loss introduces zero additional hyperparameters beyond the shared architecture.