Integrating Gradient Boosting and Generative Models: Hybrid Approach to Address Class Imbalance and Evaluation Gaps in Real-World Systems

by

Mary Lau

B.S. Computer Science and Engineering, Massachusetts Institute of Technology, 2023

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2025

Authored by: Mary Lau
Department of Electrical Engineering and Computer Science
May 9, 2025

Certified by: Amar Gupta
Research Scientist, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair
Master of Engineering Thesis Committee

Integrating Gradient Boosting and Generative Models: Hybrid Approach to Address Class Imbalance and Evaluation Gaps in Real-World Systems

by

Mary Lau

Submitted to the Department of Electrical Engineering and Computer Science
on May 9, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

Anomaly detection remains a persistent challenge in machine learning due to the extreme class imbalance, high cost of false negatives, and the need to regulate false positives in real-world settings at scale. This thesis introduces Tail-end FPR Max Recall, a business-aware evaluation framework designed for such constrained environments. Using this framework, we benchmark LightGBM—a gradient boosting method known for its computational efficiency and predictive accuracy—on an imbalanced dataset, comparing its performance against standard academic evaluation criteria. Our results demonstrate that Tail-end FPR Max Recall fills critical gaps left by standard academic criteria, providing a more realistic assessment of model performance that aims to maximize recall while enforcing a false positive rate budget. Beyond benchmarking, we propose two strategies that incorporate deep learning methods to augment the already strong performance of gradient boosting: (1) using generative models to produce synthetic minority-class samples that outperform traditional oversampling techniques, and (2) using neural embeddings to improve feature representation for anomaly detection. Together, these contributions offer a methodology for evaluating and improving anomaly detection pipelines in domains where rare, high-impact events must be detected while meeting strict operational demands.

Thesis supervisor: Amar Gupta
Title: Research Scientist

Acknowledgments

I would like to thank my thesis supervisor, Dr. Amar Gupta, for his steadfast guidance and mentorship throughout this process. When I began this journey, I was unsure whether completing a thesis was truly within my reach, but Dr. Gupta's support and experience helped me stay focused and navigate the challenges along the way. Working in his lab provided valuable opportunities for growth—both as a researcher and on a personal level. As I complete my thesis, I can confidently say that his input and encouragement were instrumental in helping me reach this milestone.

I would also like to thank the other members of this project who contributed to various aspects of the research. I am especially grateful to Sabrina Queipo, who collaborated with me on the data pre-processing and experimental setup, as well as on a related article—both of which contributed meaningfully to the development of this thesis. I would also like to thank Dr. Rafael Palacios for his thoughtful feedback on our manuscript and his guidance throughout the research and writing process.

I would like to thank the team at Itaú Unibanco—Fernando Beserra, Mariana Ferreira, Sheila Dada, Marcos Girotto, and others—for their support and collaboration throughout this project. Their engagement and feedback were valuable to the development and practical grounding of the research. I am also grateful to the other members of the lab for contributing to a collaborative and supportive environment. It was a pleasure working alongside them.

Lastly, I would like to thank my family for their unwavering support. To my mother, who encouraged me to return and complete my thesis and reminded me—firmly and often—that I could do it; and to my father, whose patience and calm support were a source of strength. I'm also grateful for their consistent reminders to take care of myself—even if it was just making sure I remembered to eat during the busiest writing days. I would also like to thank my sister, as well as my friends in the MIT and greater community, whose companionship made this journey more enjoyable and helped me stay balanced along the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the widespread digitization of practically every aspect of daily life—from tap-to-pay financial transactions to machines assisting in critical medical diagnoses to the continuous flow of network traffic—it is no longer optional, but necessary for institutions to evolve their systems into data-driven pipelines and adopt cutting-edge machine learning, lest they fall behind amid the sheer volume and velocity of big data and modern techniques. Among these applications, anomaly detection stands out as a persistent and foundational challenge: how can we identify rare, high-impact events buried within imbalanced datasets?

We are particularly motivated by anomaly detection in the context of large-scale institutions that operate under business constraints and real-time data streams. In these settings, false negatives are often extremely costly, while false positives, though tolerable in small amounts, must be kept within operational limits.

Take for instance, credit card fraud detection, where fraudulent transactions represent only a minuscule fraction of total transactions, often on the order of thousandths of a percent [1]. Failure to detect fraud (i.e. false negatives) results in financial losses for both the user and institution. On the other hand, incorrectly flagging legitimate transactions as fraud (i.e.

false positives) can inconvenience customers, generate operational overhead due to complaints and need for manual reviews, and ultimately erode user trust and brand reputation [2].

This pattern holds in other domains. In cybersecurity, systems monitor vast streams of network traffic in real time to identify potential threats. These threats are rare and often obscured within a sea of legitimate activity. They may include sudden surges in network traffic, unauthorized access attempts, and protocol violations [3]. False negatives, or failures to detect actual attacks, can lead to severe breaches, data leaks, and system-wide compromises [3]. Conversely, an excessive number of false positives—benign traffic incorrectly flagged as suspicious—can overwhelm security teams with alerts, reduce response efficiency, and lead to alert fatigue, ultimately increasing the risk of missing actual threats [4].

In the medical domain, models are trained to detect anomalies from patient records and medical imaging data [5]. False negatives in this context can be life-threatening, potentially delaying essential treatment [6]. While false positives may seem less severe, they still carry meaningful consequences—causing unnecessary stress, follow-up tests, and potentially invasive procedures [6].

From these high-stakes applications of anomaly detection across various domains, it is clear that systems must be designed to carefully balance false negatives and false positives. In practice, this means calibrating models to prioritize recall while maintaining a tightly controlled false positive rate (FPR). Yet, most academic benchmarks rely on global metrics like AUROC or F1-score, which obscure performance in the low false positive regimes that matter most in real-world deployments [7]. What's lacking is an evaluation framework that directly reflects these operational constraints and captures the cost-sensitive trade-offs at the core of anomaly detection.

Moreover, while deep learning has become the dominant data-driven paradigm in many domains due to its scalability and representational power, traditional models like Gradient Boosting remain the go-to choice for anomaly detection on structured, tabular data [8]. These models continue to outperform deep learning alternatives in both accuracy and computational

efficiency [8]. This raises a compelling question: Can we leverage deep learning to enhance traditional models like Gradient Boosting for anomaly detection?

This thesis explores both of these gaps—in evaluation and in modeling. We propose a business-aligned evaluation framework tailored to constrained operational settings, and we investigate how deep learning techniques can augment Gradient Boosting through synthetic data generation and neural feature embeddings. Ultimately, we aim to bridge the divide between academic research and real-world deployment in high-stakes anomaly detection systems.

1.2 Problem Statement

In this thesis, we focus on anomaly detection in environments characterized by three key challenges:

- **Extreme class imbalance**, where the proportion of anomalies is significantly smaller than that of the majority class;
- **Asymmetric error costs**, where false negatives and false positives have different costs and implications;
- **Business-aligned operating constraints**, where as many false negatives as possible must be detected, but excessive false positives can create operational overhead and lead to user dissatisfaction.

Despite the prevalence of these conditions in various real-world domains—such as fraud detection, cybersecurity, and medical diagnostics—there is no widely adopted evaluation framework in academic research that captures model performance under these specific operational constraints. Most academic metrics, such as AUC-ROC and F1-score, reflect aggregate performance and fail to capture model behavior in the low false positive regimes where anomaly detection systems are typically deployed. At the modeling level, deep

learning has shown promise in high-dimensional and unstructured data domains, but remains underutilized in tabular anomaly detection. In practice, Gradient Boosting remains the dominant approach due to its efficiency and strong empirical performance. However, it remains unclear how deep learning methods—particularly generative models and neural embeddings—might be integrated to enhance Gradient Boosting, especially in addressing challenges like class imbalance and feature representation.

This thesis addresses both of these gaps: (1) the absence of a business-aligned evaluation framework tailored to real-world anomaly detection, and (2) the unexplored potential of using deep learning to improve traditional Gradient Boosting methods in structured data settings.

1.3 Contributions

This thesis makes the following contributions:

1. Tail-end FPR Max Recall: A Business-Aligned Evaluation Framework

We formalize an evaluation framework that captures the real-world constraints faced by high-stakes anomaly detection systems. Rather than focusing on average-case performance across the full FPR-recall curve, **Tail-end FPR Max Recall** emphasizes recall at low false positive rates—a metric that directly aligns with how institutions assess model quality and make deployment decisions in practice.

2. Comparison of Tail-end FPR Max Recall Framework with Standard Academic Metrics

We evaluate LightGBM on a tabular anomaly detection dataset using both Tail-end FPR Max Recall and standard metrics such as AUC-ROC and F1-score. Our results show that Tail-end

FPR Max Recall more effectively captures model performance under low false positive rate constraints, revealing critical differences that traditional, aggregate metrics often obscure.

3. Deep Learning-enhanced Gradient Boosting

We explore two methods for augmenting Gradient Boosting with deep learning techniques to address key challenges such as class imbalance and limited feature expressiveness, with the goal of improving downstream classification.

- **Synthetic Data Generation:** Leveraging generative models to produce realistic minority-class samples to increase the minority class representation.
- **Neural Feature Embeddings:** Using deep learning to learn latent representations that can enrich tabular input data and feature representation.

1.4 Thesis Outline

In this thesis, we begin by surveying the literature on anomaly detection in imbalanced settings, with a focus on evaluation metrics, Gradient Boosting methods, and recent advances in deep learning. We also review techniques for synthetic data generation, neural embeddings for tabular data, and hybrid modeling strategies.

We then formalize the anomaly detection problem in high-stakes, real-world environments—such as credit card fraud detection—characterized by extreme class imbalance, asymmetric error costs, and operational constraints. Motivated by the limitations of conventional metrics like AUC-ROC and F1-score, we introduce Tail-end FPR Max Recall, a business-aligned evaluation framework that emphasizes recall under low false positive rate conditions.

Using this framework, we benchmark the performance of Light Gradient Boosting model on an imabalanced tabular dataset, comparing results under both standard and Tail-end FPR Max Recall metrics.

Building on these findings, we explore hybrid approaches that enhance Gradient Boosting with deep learning components—specifically, by using generative models to synthesize minority-class data and by incorporating neural embeddings to improve feature representation.

Finally, we present a detailed evaluation of both standalone and hybrid methods under our proposed framework, summarize key insights, and discuss implications and future work.

Chapter 2

Related Work

2.1 Anomaly Detection in Imbalanced Settings

Anomaly detection has been extensively studied using a variety of machine learning techniques, ranging from traditional statistical models to advanced deep learning architectures [9]. Due to the inherent class imbalance in anomaly detection tasks, many methods aim address imbalance via either data-level or algorithm-level methods [10].

2.1.1 Data-Level

Data-level methods adjust the training dataset's distribution to be more balanced between classes. This may involve increasing the size of the minority class, removing samples from the majority class, or both. For instance, Random Oversampling [11] increases the number of minority class instances by randomly duplicating existing minority samples with replacement. SMOTE (Synthetic Minority Over-sampling Technique) [12] and its variants (e.g., ADASYN) [13] use interpolation and perturbation of existing samples to create new minority-class points. These methods work to reduce the class imbalance within the dataset, prior to the model or algorithm selection stage.

2.1.2 Algorithm-Level

Traditional algorithm-level methods include Logistic Regression, Support Vector Machines (SVMs), Decision Trees, Naive Bayes Classifiers, Nearest Neighbors, and Random Forests. These are all classical approaches in this domain. These methods are valued for their interpretability and computational efficiency, but they often face challenges in handling imbalanced datasets, high-dimensional feature spaces, and evolving anomaly patterns [14, 15].

Another common algorithm-level approach is making modifications to the model's loss function. One example is adjusting the loss function to increase the penalty for misclassifying positive instances. Many machine learning libraries implement this through parameters like class_weight and sample_weight [16], allowing models to emphasize underrepresented classes during training.

2.2 Gradient Boosting in Tabular Anomaly Detection

In practice, Gradient Boosting is the dominant choice of model used for handling structured tabular datasets. Specifically, XGBoost, CatBoost, and LightGBM are among the Gradient Boosting methods that are widely adopted due to their strong performance in anomaly detection tasks [17–19].

Central to our study, Microsoft's LightGBM [20] is a gradient-boosting decision tree framework that incorporates two key innovations: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). These techniques enable efficient handling of large datasets and high-dimensional feature spaces, which are typical in financial institutions, without compromising performance.

Previous research [21] applied LightGBM to the IEEE fraud detection dataset [22]. Similarly, another study applied LightGBM to the MLG dataset, which is the dataset used in our study [23].

2.3 Deep Learning for Anomaly Detection

Deep learning methods, such as Convolutional Neural Networks (CNNs), Graph Neural Networks (GNNs), Autoencoders, and Extreme Learning Machines (ELMs), have also been employed for anomaly detection. They are known for their ability to excel at automatic feature extraction and recognizing complex patterns within large datasets [24–26]. However, these methods are often criticized for their "black-box" nature and may require substantial computational resources for both training and inference.

For datasets that are sequential in nature, Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs) have been widely used to model sequences and detect anomalies based on historical data [27, 28]. More recently, Transformer-based architectures have been used for anomaly detection, leveraging transfer learning to detect anomaly patterns across datasets [29, 30].

Despite the groundbreaking success of deep learning models across a variety of language tasks (e.g., natural language processing, machine translation, sentiment analysis) [31–33], tree-based ensemble methods remain a competitive choice for handling structured tabular datasets [34].

2.4 Generative Approaches

Synthetic Data Generation

Synthetic data generation is a method used to address class imbalance in domains like financial fraud detection and deepfake detection, where data is often scarce or sensitive. Methods such as TVAE (Tabular Variational Autoencoder) and CTGAN (Conditional Tabular GAN) [35] have been tailored for modeling tabular data, generating synthetic data that captures the underlying patterns of real-world datasets and improve minority class representation.

2.4.1 Embeddings Generation

Another promising approach involves leveraging generative models—such as neural networks, autoencoders, and Transformer-based architectures like TabTransformer [36]—to capture meaningful, lower-dimensional structures from tabular data. These embeddings, having learned compact latent representations, can be used to enhance model performance on downstream classification tasks like anomaly detection [37, 38].

2.5 Evaluation Metrics for Anomaly Detection

Using model accuracy, while intuitive, can be misleading in imbalanced datasets, as demonstrated by the fact that predicting all transactions as non-anomaly can still yield over 99% accuracy. Another common evaluation tool for class-imbalanced tasks is generating a confusion matrix, which consists of four key components: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Other metrics such as accuracy, precision, recall, and AUC are derived from these components. There is also ROC (Receiver Operating Curve), which plots True Negative Rate with respect to False Positive Rate. AUROC is the total area under the ROC curve. Partial AUC (pAUC) is similar, but only calculates a portion of the ROC curve. Most papers in anomaly detection use ROC and AUROC to evaluate their methods, but not many use Partial AUC [39].

Chapter 3

Design and Methodology

3.1 Problem Formulation

3.1.1 Notation

We consider a binary anomaly detection problem in a structured data setting. The objective is to identify rare and potentially harmful instances—referred to as anomalies—within a large population of normal data points.

Formally, let the dataset be defined as:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}, \quad x_i \in R^d,\ y_i \in \{0, 1\}$$

where:

- x_i represents the feature vector of the i-th data sample instance,
- $y_i = 1$ indicates the instance is an anomaly,
- $y_i = 0$ indicates the instance is normal (non-anomalous),
- n is the number of samples in the dataset, and d is the number of features per sample.

$f : R^d \rightarrow R$ represents a function that outputs an anomaly score for each input. This score is typically interpreted as the likelihood that an instance is anomalous. A threshold τ is used to determine which binary prediction to assign to the input:

$$\hat{y}_i = \begin{cases} 1 & \text{if } f(x_i) \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

where:

- $\hat{y}_i$ is the predicted label of x_i
- $f(x_i)$ is the model's anomaly score for x_i
- τ is a threshold that determines the decision boundary,

What we seek to do in this anomaly detection problem is to learn the best model f and best operating threshold τ that can detect anomalies with high model performance under practical constraints. These specific constraints include consideration of the rate of true negatives and rate of false positives (metrics that will be elaborated in Section 3.2).

3.1.2 Class Imbalance

Anomaly detection problems are frequently defined by **extreme class imbalance**, where the proportion of anomalies is significantly smaller than that of the majority class. This imbalance can be quantified using the *imbalance ratio* (IR), defined as [40]:

$$\text{IR} = \frac{N_{maj}}{N_{min}}$$

where:

- $N_{maj} = |\{i : y_i = 0\}|$ is the number of normal (majority class) data samples, and
- $N_{min} = |\{i : y_i = 1\}|$ is the number of anomalous (minority class) data samples.

In many real-world applications, this ratio can be extremely large— on the order of 1000:1, or greater. Such imbalance makes it challenging for models to learn meaningful patterns from the minority class, often resulting in classifiers that bias their predictions toward the majority class.

To address this, specialized modeling strategies, including class-weighted loss functions, targeted sampling techniques, and synthetic data generation, are commonly employed to address class imbalance on an algorithm-level and data-level. However, improving model training is only part of the solution; more specialized evaluation strategies are also needed.

3.1.3 Standard Evaluation Metrics

To evaluate binary anomaly detection models, researchers commonly rely on standard metrics derived from following four key quantities:

- **True Positives (TP)**: the number of anomalous instances correctly predicted as anomalies.
- **False Positives (FP)**: the number of normal instances incorrectly predicted as anomalies.
- **True Negatives (TN)**: the number of normal instances correctly predicted as normal.
- **False Negatives (FN)**: the number of anomalous instances incorrectly predicted as normal.

These four counts are typically visualized as a confusion matrix (see Figure 3.1). The rows of the confusion matrix represent the actual classes (negative label if normal and positive label if anomalous), while the columns represent the predicted classes. Each cell in the matrix indicates the number of instances that fall into a specific prediction–ground truth combination, allowing for clear display of the number of true positives, false positives, true negatives, and false negatives.

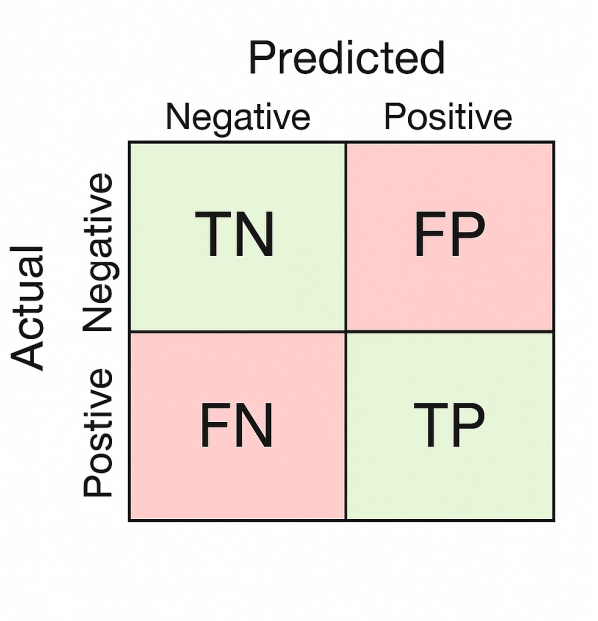

Figure 3.1: An example confusion matrix illustrating the four different actual-predicted outcomes in binary anomaly detection. Incorrect prediction quantities are colored in red, correct in green.

From these four quantities of the confusion matrix, the following standard metrics are commonly derived and reported in academic literature:

Recall (True Positive Rate)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Measures the proportion of actual anomalies that are correctly detected.

Precision

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Measures the proportion of predicted anomalies that are actually anomalous.

False Positive Rate (FPR)

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

Measures the proportion of normal instances that are incorrectly flagged as anomalies.

F1 Score

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Provides the harmonic mean of precision and recall.

Area Under the ROC Curve (AUROC) The ROC Curve plots true negative rate (i.e. recall) on the y-axis with respect to false positive rate (i.e. FPR) on the x-axis. AUROC is the total area under this cover across the entire FPR x-axis range (from 0% to 100%). Many studies report AUROC to summarize model performance, using this value to compare between different classifiers.

3.1.4 Asymmetric Error Costs

While standard evaluation metrics such as precision, recall, and AUROC provide useful aggregate measures of model performance, they implicitly equate the cost of all prediction error types. In many real-world anomaly detection tasks, however, false negatives versus false positives carry significantly different consequences.

In some applications, the cost of a false negative may be higher than the cost of a false positive. In other cases, false positives may be more problematic. The relative importance of these error types depends on the domain and operational constraints, highlighting the need for evaluation metrics that account for asymmetric costs and real-world trade-offs—something that standard metrics do not always capture effectively.

In the context of this paper, which selects a credit card fraud detection dataset for its use case, our models must be calibrated to prioritize recall (i.e., minimize false negatives) while keeping the false positive rate within acceptable operational bounds. With this use case in mind, we introduce an evaluation framework—*Tail-end FPR Max Recall*—designed to better capture model performance under these real-world trade-offs.

3.2 Tail-end FPR Max Recall: A Business-Aligned Evaluation Framework

We observe in our literature review that the standard evaluation approach is to report the recall, precision, F1-score, aggregated across the entire test dataset, as well as the total AUROC across the entire FPR x-axis range. These metrics are good, but are not sufficient to account for the specific operating regimes and cost structures of real-world anomaly detection systems.

In the practical field of financial fraud detection, financial institutions are often required to maintain false positive rates (FPR) below a domain-specific threshold ϵ. These systems must not only be effective at detecting the rare occurrences of actual fraud but must also avoid generating excessive false positives (i.e., flagging legitimate customer transactions as fraudulent) which can result in customer inconvenience, operational overhead, and increased support costs.

Definition

With these constraints and cost structures in mind, we define the **Tail-end FPR Max Recall** as the maximum achievable recall subject to an upper bound on the false positive rate:

$$\text{Tail-end FPR Max Recall} = \max_{\tau} \{\text{Recall}(\tau) \mid \text{FPR}(\tau) \leq \epsilon\}$$

where:

- τ is a decision threshold applied to the model's anomaly score function,
- $\text{Recall}(\tau) = \frac{\text{TP}(\tau)}{\text{TP}(\tau)+\text{FN}(\tau)}$ is recall at threshold τ,
- $\text{FPR}(\tau) = \frac{\text{FP}(\tau)}{\text{FP}(\tau)+\text{TN}(\tau)}$ is the false positive rate at τ,

- ϵ is a predefined FPR budget (i.e. the system tolerates at most $\epsilon\%$ of all normal transactions being flagged as fraud).

This formulation aligns with how anomaly detection models are assessed in production: models are selected or tuned not based on their overall ROC curve, but on their ability to deliver the highest possible recall under an operationally acceptable false positive rate. This region—typically the far left "tail" of the ROC curve—is rarely emphasized by traditional metrics, which average performance across all possible FPR values.

Tail-end FPR Max Recall makes this trade-off explicit, providing a more realistic and actionable measure of model effectiveness in high-precision, cost-sensitive environments. It is used throughout our benchmarking and hybrid modeling experiments to compare models under business-aligned conditions.

3.3 Integrating Deep Learning into Gradient Boosting

3.3.1 Gradient Boosting Models

LightGBM LightGBM is a gradient boosting framework based on decision trees, designed for speed and efficiency in large-scale data scenarios [16]. It uses histogram-based training and a leaf-wise tree growth strategy, which helps achieve faster convergence and improve accuracy compared to traditional level-wise boosting approaches. Additionally, LightGBM natively handles missing values and categorical features, further simplifying preprocessing for structured data.

We adopt LightGBM as our reference model due to its widespread use in industry and its consistently strong empirical performance on tabular datasets. All model comparisons throughout this thesis are made relative to this baseline.

3.3.2 Deep Learning Models

Autoencoder Autoencoders are neural networks designed to learn compact representations of input data by reconstructing the input from a lower-dimensional encoding [41, 42]. They consist of two components: an encoder that maps the input $x \in R^d$ to a latent representation $z \in R^k$ (with $k < d$), and a decoder that attempts to reconstruct the input from this latent space, producing $\hat{x} \in R^d$:

$$z = f_{\text{enc}}(x), \quad \hat{x} = f_{\text{dec}}(z)$$

The model is trained to minimize reconstruction loss between the input and output, typically using mean squared error (MSE):

$$\mathcal{L}_{\text{recon}} = \|x - \hat{x}\|^2$$

In anomaly detection, the autoencoder is trained primarily on normal (non-anomalous) data. As such, the autoencoder should be less effective at reconstructing anomalous instances that deviate from the training distribution, and the reconstruction loss can be used as an anomaly score[43, 44]:

$$s(x) = \|x - \hat{x}\|^2$$

A threshold τ is applied to $s(x)$ to determine a binary anomaly prediction. Inputs with reconstruction error above τ would be flagged as anomalies.

Variational Autoencoder (VAE) The variational autoencoder (VAE) extends the standard autoencoder by introducing a probabilistic latent space [45]. Instead of encoding each input to a fixed point z, the encoder learns the parameters of a distribution over latent variables. For input x, the encoder outputs the mean and standard deviation of a Gaussian distribution:

$$\mu, \sigma = f_{\text{enc}}(x), \quad z \sim \mathcal{N}(\mu, \text{diag}(\sigma^2))$$

The decoder then reconstructs the input from a sample z drawn from this distribution:

$$\hat{x} = f_{\text{dec}}(z)$$

The VAE is trained to minimize a loss function that combines reconstruction error with a regularization term [46]. The regularization term ensures that the learned posterior remains close to a standard normal prior in order to help encourage smoothness and continuity in the latent space:

$$\mathcal{L}_{\text{VAE}} = E_{q(z|x)}[\|x - \hat{x}\|^2] + \beta \cdot D_{\text{KL}}(q(z|x) \parallel p(z))$$

where D_{KL} is the Kullback–Leibler divergence, and β is a scaling factor controlling the regularization strength [45, 47].

Tabular Variational Autoencoder (TVAE) Unlike the VAE—which assumes continuous inputs and Gaussian decoders—TVAE customizes its encoder–decoder pipeline for mixed-type tabular data [35]. In pre-processing, TVAE transforms continuous columns into Gaussian mixtures and one-hot encodes categorical columns. In the decoder, TVAE adds two specialized heads: a regression head for continuous outputs and Gumbel-Softmax heads for both mixture-component indicators and categorical variables [35, 48]. This design allows it to faithfully capture the heterogeneous nature of tabular data. Due to its generative structure and handling of mixed-type data, TVAE is particularly well suited for anomaly detection tasks in tabular datasets.

3.3.3 Hybrid Approaches: Enhancing Gradient Boosting with Deep Generative Models

While gradient boosting remains the dominant method for anomaly detection in tabular data, we explore in this study how deep learning can be used to augment its performance. We propose two hybrid strategies: (1) synthetic data generation to address class imbalance, and

(2) neural embeddings to enrich feature representations.

(1) Synthetic Data Generation To mitigate the class imbalance problem, we use a deep generative model—a tabular variational autoencoder (TVAE)—trained exclusively on minority-class (anomalous) instances to synthesize new, realistic fraud samples. These generated samples are added to the original training dataset to rebalance class distribution. The rebalanced dataset is then used to train the Gradient Boosting (LightGBM) classifer. This pipeline is illustrated in Figure 3.2:

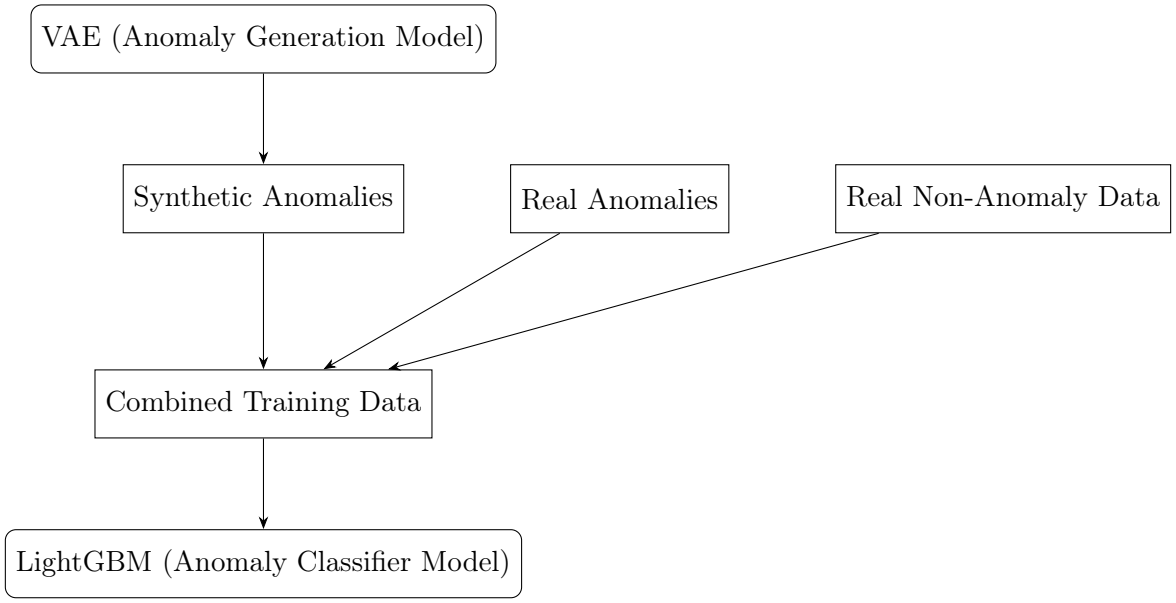

Figure 3.2: Synthetic anomaly samples generated by TVAE are combined with real anomalies and majority-class training data to train a LightGBM classifier. Model nodes are rounded; data nodes are not.

This approach aims to improve model performance by generating new minority class samples that are realistic and similar to the original instances, yet distinct enough to promote better generalization.

(2) Neural Feature Embeddings We also explore the use of deep generative models to enhance input representations for tabular data. We train a neural network on the dataset's

feature space and extract the activations from a hidden layer as a learned embedding:

$$z = f_{\text{embed}}(x)$$

These dense, non-linear embeddings z are concatenated with the original features and passed onto the LightGBM model, as illustrated by Figure 3.3:

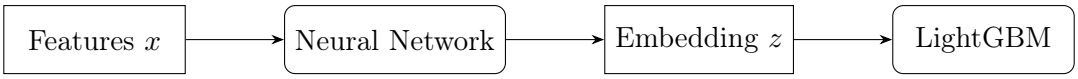

Figure 3.3: Neural embeddings extracted from a hidden layer are used as inputs to LightGBM.

This approach aims to improve model performance by enriching the input space with learned representations that capture non-linear feature interactions. By combining these embeddings with the original features, the model gains access to both raw and abstracted views of the data, potentially enhancing its ability to distinguish between classes.

In this study, we test these hybrid approaches to investigate whether deep learning can enhance traditional classifiers, targeting the challenges of imbalance and feature complexity, ultimately improving anomaly detection performance in a real-world financial dataset.

3.4 Data and Experimental Setup

3.4.1 Dataset Description

During the process of searching for a suitable financial fraud anomaly dataset, we found multiple dataset options, both synthetic and realistic. Although there are quite a few papers that use or propose synthetically generated datasets [49, 50], we ultimately decided on a realistic dataset. Realistic and publically available datasets are much rarer, as financial institutions are very strict about disclosing transaction data, due to user confidentiality and privacy regulations. The few realistic financial fraud datasets that are publically available [22, 51] come with heavily anonymized or obfuscated features. For instance, the IEEE dataset

includes over 300 features per sample, but fewer than 10% of these features are clearly labeled, while the rest have masked names, making their semantic meaning unclear [22].

For our study, we decide on the MLG dataset [51], a credit card transaction dataset collected from European cardholders in 2013, anonymized and released by the French fintech company Worldline. The MLG dataset spans two days and is highly imbalanced, with 492 fraudulent transactions out of 284,807 total transactions (making it 0.172% fraud). It contains 31 numerical features: `Time`, `Amount`, 28 anonymized PCA-transformed numerical variables (`V1`, `V2`, ..., `V28`), and the binary fraud label.

We removed the `Time` column and standardized the `Amount` column, as these preprocessing steps were found to improve model performance. An overview of the preprocessed data is presented in Table 3.1.

Table 3.1: Feature Schema of the MLG Dataset (30 Total Features)

Feature	**Description**	**Data Type**
V1–V28	Anonymized Features	Numeric
Std(Amount)	Standardized Transaction Amount	Numeric
Fraud	Indicator of Fraud	Binary (0/1)

3.4.2 Preprocessing and Training Procedure

We apply a time-series split, using the first 80% of the data for training and the most recent 20% for testing. The training split contains 0.183% fraud, with 417 fraud instances and 227,428 non-fraud instances, totaling 227,845 samples. The test split contains 0.13% fraud, with 75 fraud instances and 56,887 non-fraud instances, totaling 56,962 samples.

To select optimal hyperparameters, we use the Optuna package [52] with 3-fold cross-validation. The final configuration for the LightGBM model is shown in Table 3.2.

Table 3.2: Final Hyperparameters for the LightGBM Model

Hyperparameter	**Description**	**Value**
NumLeaves	Maximum number of leaves in one tree	31
LearningRate	Step size	0.0190
FeatureFraction	Fraction of features used in building trees	0.7320
BaggingFraction	Fraction of data used for bagging	0.7547
MinDataInLeaf	Minimum data points required in a leaf node	30
LambdaLOne	L1 regularization term on weights	1.4452
LambdaLTwo	L2 regularization term on weights	0.4575

Chapter 4

Results

4.1 Evaluating Quality of TVAE-Generated Synthetic Data

We train a TVAE on the original fraud data in the training set to generate synthetic fraud samples. Figure 4.1 visualizes the resulting distribution, with synthetic fraud data shown in yellow, original fraud in red, and original non-fraud in green. To illustrate how well the synthetic samples capture the underlying distribution of real fraud cases, we highlight the top three feature pairs most correlated with the fraud label. The left column shows the original dataset, the middle adds the synthetic samples, and the right greys out the original data to better isolate the synthetic distribution.

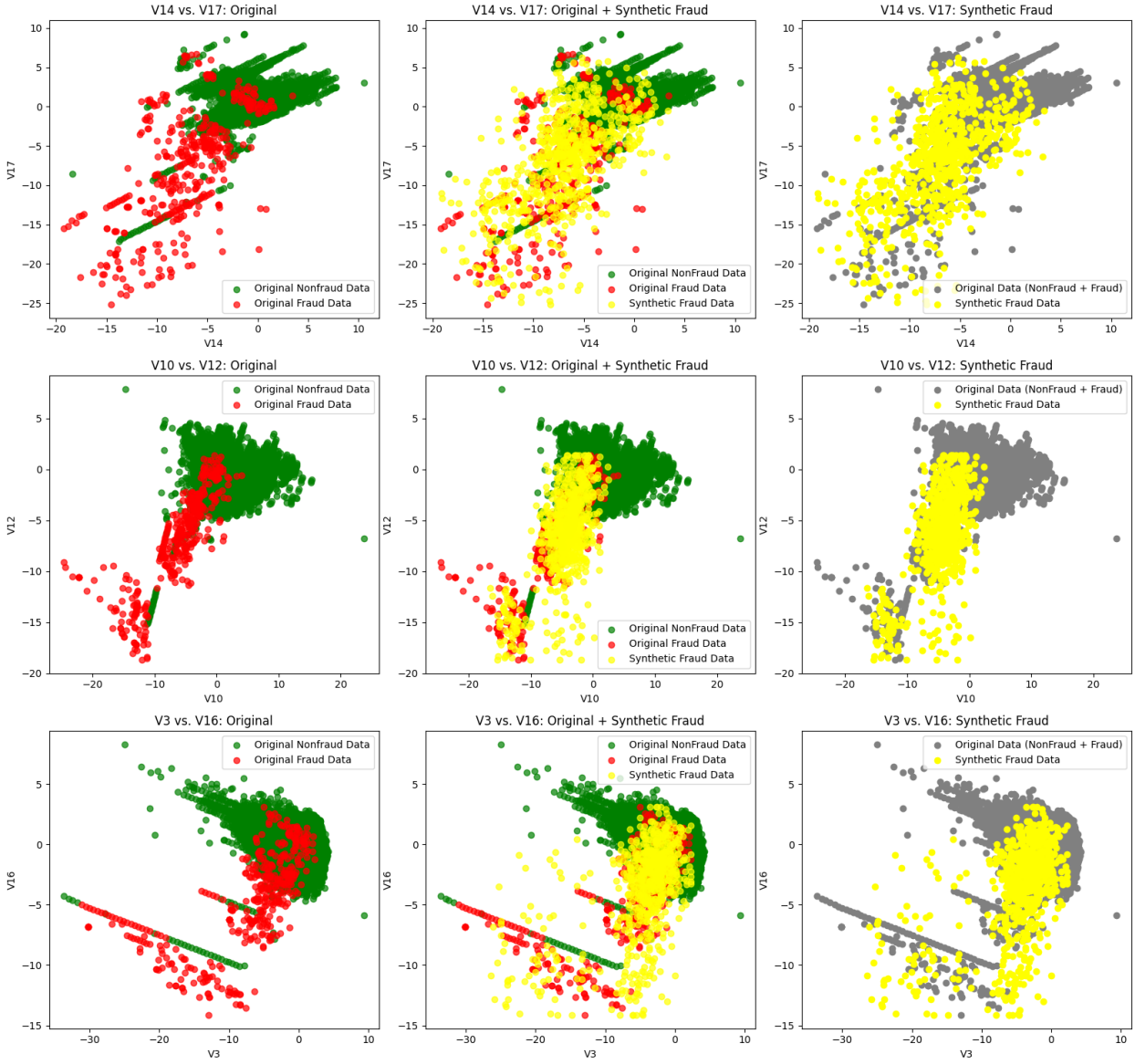

Figure 4.1: Top three feature pairs most correlated with the fraud label. Green points are original non-fraud samples, red points are original fraud, and yellow points are synthetic fraud generated by a TVAE trained on the original fraud data. Left: original dataset. Middle: original + synthetic. Right: synthetic only (original data greyed out).

This visualization helps us assess how well the synthetic data captures the distribution of the original fraud instances. In Figure 4.1, we observe that the yellow synthetic fraud data points remain localized within the region of the original fraud data but do not perfectly replicate them. Instead, the TVAE introduces slight perturbations, producing samples that are similar yet distinct—an effect that can support better model generalization. Moreover, in terms of decision boundaries between fraud and non-fraud transactions, the synthetic points largely remain on the correct side and cluster near true fraud instances. From this, we qualitatively assess that the generated synthetic data is of good quality.

We hypothesize that training the VAE's encoder to learn a latent representation of the fraud class, and using the decoder to introduce controlled perturbations, enables the generation of synthetic fraud samples that help classifiers generalize to rare or more borderline cases. Thus, TVAE-generated data is expected to be more effective than random oversampling, as it introduces realistic variation rather than simply duplicating existing fraud instances, thereby improving model robustness to rare and diverse fraud patterns.

4.2 Applying the Tail-end FPR Max Recall Framework Compared to Standard Academic Evaluation

4.2.1 Threshold Selection under FPR Budgets Improves Model Performance at Scale

Using LightGBM's default prediction settings and scikit-learn's evaluation metrics (recall, precision, FPR, confusion matrix), we generate the confusion matrix shown in Figure 4.2 (top) [16, 53]. This evaluation approach is the standard practice adopted in nearly all related studies we reviewed. Based on this standard academic evaluation, the model achieves a strong recall of 82.67% and a high precision of 99.81%—metrics that would be considered excellent in most academic fraud detection literature. However, the false positive rate (FPR) stands at

0.19%— a rate that would be problematic in a banking environment processing millions of transactions per day. At that scale, a 0.19% FPR would result in thousands of legitimate transactions being incorrectly flagged as fraud per day, leading to signifcant operational strain, customer dissatisfaction, and direct financial losses for the institution.

Thus, the importance of strictly controlling the false positive rate (FPR) while achieving high recall cannot be overstated. To do this, our proposed framework first sets an FPR budget—the maximum percentage of false positives the system is allowed to tolerate. For this example, we set the FPR budget to 0.05%, meaning that for every one million transactions, no more than 500 legitimate transactions can be incorrectly flagged as fraud (a substantial improvement). With this budget defined, we then scan all possible thresholds of the anomaly score and identify the highest threshold that still ensures the FPR remains within the limit. In this example, that threshold is 787 (on an anomaly score scale from 0 to 1000).

Using this new threshold, we generate the confusion matrix shown at the bottom of Figure 4.2, confirming that the FPR is indeed 0.05%, as specified. In other words, the number of false positives has been reduced by 74%, with only a 3.3% drop in recall. Precision also improves slightly, increasing by 0.14%. Notably, the number of real transactions misclassified as fraud falls from 108 to 28, while the number of fraud transactions incorrectly flagged as real remains nearly unchanged (15 instead of 13).

The key takeaway is that while performance gains of a few tenths or hundredths of a percent may seem negligible in academic settings, they can translate into thousands of transactions in real-world financial companies that handle millions of transactions daily—leading to significant operational and financial impact.

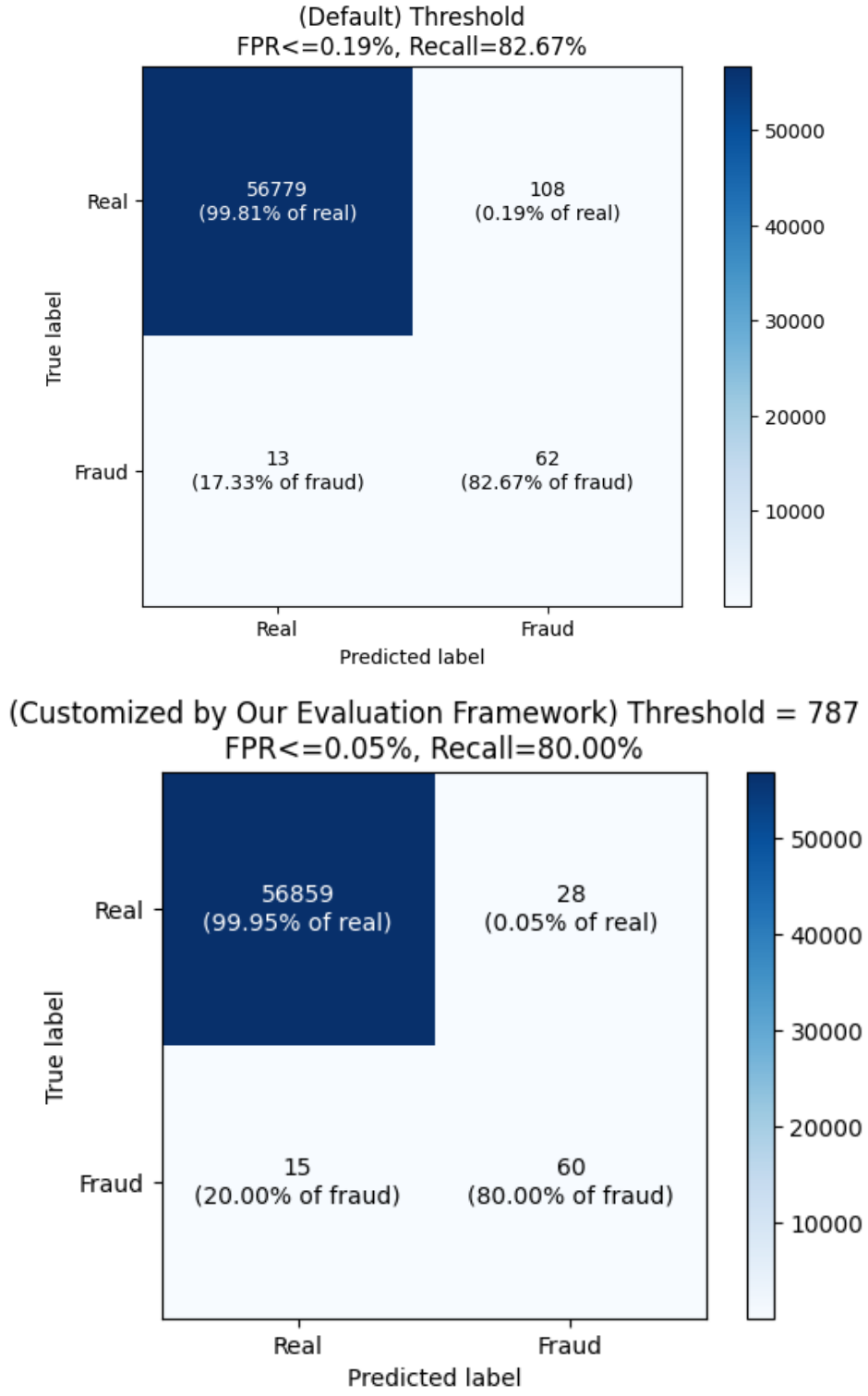

Figure 4.2: Confusion matrices comparing model performance using standard academic evaluation approach with default threshold (top) and the proposed Threshold Selection under FPR Budget (part of our evaluation framework) (bottom).

4.2.2 Augmenting with TVAE-Generated Synthetic Data Outperforms ROS and Original Data

In Section 4.1, we qualitatively assessed the quality of our TVAE-Generated synthetic data. In Section 4.2.1, we described our process for threshold selection under a set FPR budget. Using confusion matrices (standard academic evaluation tool), we confirmed that implementing this step leads to improvement in FPR (and Precision) that would be indispensable to financial real-world systems at scale.

Now, in this section, we will extensively evaluate the impact of augmenting our dataset with TVAE-generated synthetic data, using our proposed evaluation framework (Tail-end FPR Max Recall) to compare between the various anomaly (fraud) classifiers trained on differently augmented dataset variants. Moreover, for each classifier that we train, we will carry out threshold selection under a set FPR budget, as demonstrated in Section 4.2.1– this is actually a crucial step of the Tail-End FPR-Max Recall evaluation framework.

Experiment Details

For our experiments, we have three variants of the training data set: (1) the original training dataset which has 0.172% fraud, (2) the training dataset enhanced with Random Oversampling (ROS) to be 1% fraud, and (3) the original training dataset enhanced with synthetic data generated by TVAE to be 1% fraud.

For each variant of the dataset, we use Optuna to select optimal hyperparameters [54]. Then, for each dataset variant, we train LightGBM across 50 different seeds.

Experiment Results Analysis

From each of the 50 experiment seeds, we record the recall values across a range of FPR budgets, ranging from 0% to 0.10%, which we are able to determine using the Threshold Selection process described in 4.2.1. We report the average of the recall experimental results

across the 50 seeds in Table 4.1, as well as 95% confidence intervals so that we can assess whether observed performance differences are statistically significant

Table 4.1: Performance metrics at key FPR thresholds and corresponding AUC values. Values are averaged over 50 seeds with 95% confidence intervals. Boxed results are highlighted for further discussion in the following text.

FPR (%)	Recall (%) – Original	Recall (%) – 1% (ROS)	Recall (%) – 1% (VAE)
0	37.49	17.57	2.43
0.01	70.48	72.35	73.57
0.02	76.11	76.51	76.96
0.03	77.12	77.65	78.72
0.04	77.52	78.69	79.73
0.05	78.21	79.20	80.61
0.06	78.77	79.76	80.88
0.07	79.17	80.21	80.93
0.08	79.49	80.67	81.17
0.09	79.79	80.93	81.30
0.10	79.89	81.04	81.44
AUC@FPR=[0,100%]	0.982 (CI: 0.981, 0.983)	0.973 (CI: 0.971, 0.975)	0.986 (CI: 0.985, 0.987)
AUC@FPR=[0,0.1%]	7.55e-4 (CI: 7.53e-4, 7.58e-4)	7.55e-4 (CI: 7.51e-4, 7.59e-4)	7.55e-4 (CI: 7.53e-4, 7.57e-4)
AUC@FPR=[0,0.01%]	5.40e-5 (CI: 5.21e-5, 5.59e-5)	4.50e-5 (CI: 4.13e-5, 4.87e-5)	3.80e-5 (CI: 3.63e-5, 3.97e-5)
AUC@FPR=[0.01%,0.1%]	7.01e-4 (CI: 7.00e-4, 7.02e-4)	7.10e-4 (CI: 7.08e-4, 7.12e-4)	7.18e-4 (CI: 7.17e-4, 7.19e-4)

In addition to reporting recall across FPR budgets, we also evaluate overall classifier performance by plotting the ROC curve for each dataset variant. Each curve is computed by averaging the ROC curves across 50 experimental seeds, and the resulting plot—shown in Figure 4.3—spans the full FPR range. The legend of the figure reports the Total AUC (Area Under the Curve) for each variant. This evaluation approach—plotting the full ROC curve and summarizing performance using the total area under the curve (AUROC)—is commonly used in many of the academic papers we reviewed.

Looking at the top plot of Figure 4.3, which displays the full recall range (0% to 100%), we see that it is difficult to distinguish differences between the curves, particularly in the high-recall region. To address this, the bottom plot provides a zoomed-in view of the 80% to 100% recall range. In this region, we observe that classifiers trained on data augmented with TVAE-generated synthetic fraud tends to have higher recall across the entire FPR ranges than classifiers trained on the original Dataset or the original Dataset with Random Oversampling (ROS).

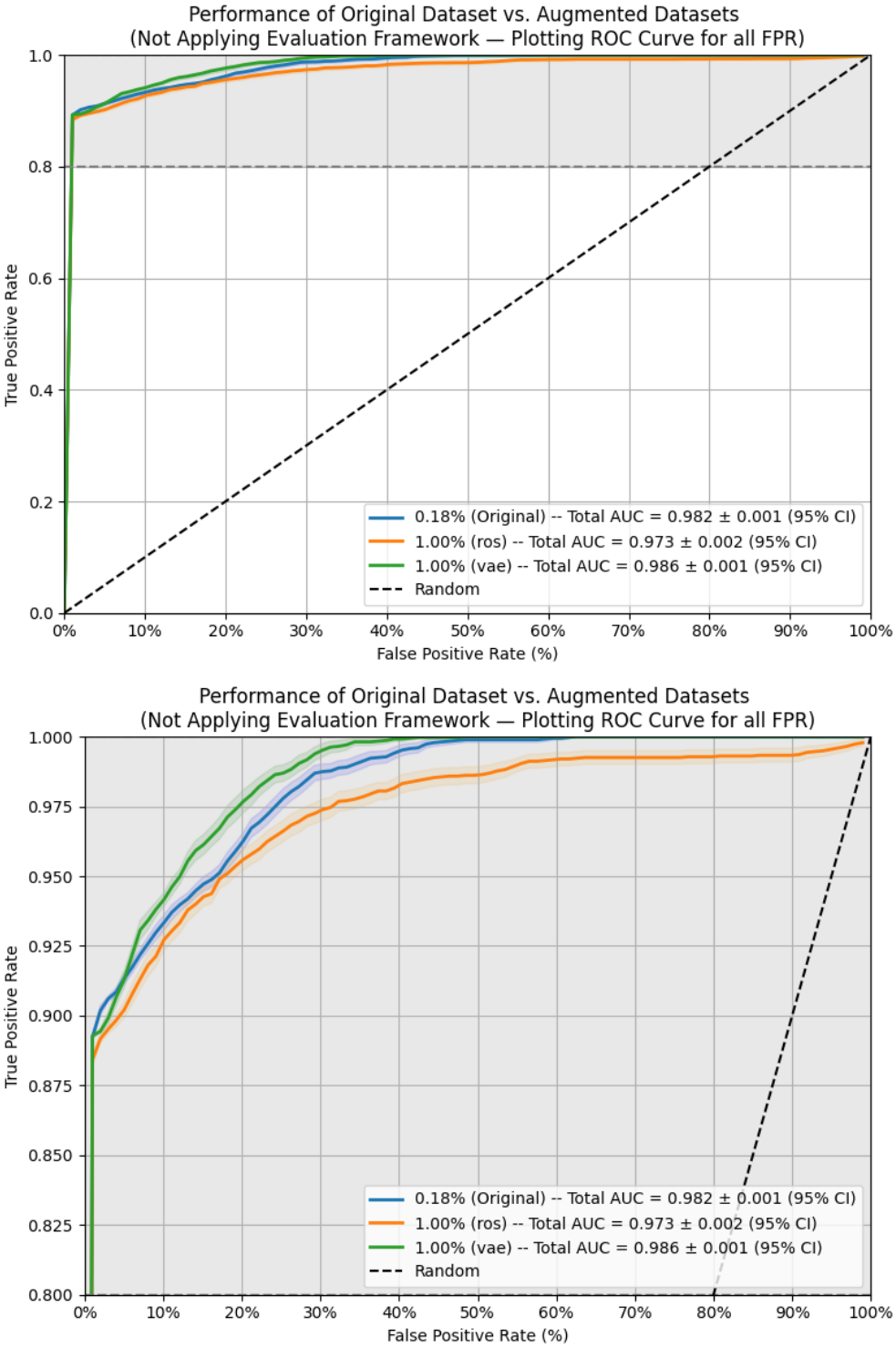

Figure 4.3: ROC curves for all classifiers across the full range of false positive rates (FPR). The top plot shows the full ROC curve. The bottom plot zooms into the high-recall region (80%–100%) to better show performance differences.

From this, we conclude that classifiers augmented with TVAE-generated data (green) consistently outperform those trained on the other dataset variants—namely, the original dataset (blue) and the ROS-augmented dataset (orange)—across most of the FPR range. Interestingly, it seems that ROS-augmented classifiers tend to perform worse than classifiers trained on the baseline dataset across much of the curve, suggesting that sampling methods do not necessarily improve performance and may, in some cases, degrade it.

However, we are not satisfied with this evaluation alone. As previously discussed, in large-scale systems, it is critical to ensure that the false positive rate (FPR) falls within an operationally acceptable budget. Even if classifiers achieve high recall, the results become far less useful if they occur at FPR levels of 10%, 5%, or even 1% (1% of one million transactions still means 10,000 legitimate transactions being incorrectly flagged in production).

This is where the Tail-end FPR Max Recall framework comes in. By setting a business-acceptable FPR budget—for example, allowing at most 0.1% of normal transactions to be falsely flagged—we can determine the corresponding anomaly score threshold and inspect the model's recall at that point. This portion of the ROC curve, located at the left tail end of the FPR x-axis, represents the most operationally realistic evaluation for deployment. In Figure 4.4, we zoom in on this region of the ROC curve, focusing specifically on FPR values less than 0.1%. We also record in the legend the partial AUC of each roc curve for FPR $\leq 0.1\%$.

We observe in this plot for FPR less than 0.1%, there is point where the classifiers per dataset variant seems to diverge in behavior, this being the region where FPR $\leq 0.01\%$ (ultra-low FPR), which is greyed out in the plot, and secondly, the region where FPR $> 0.01\%$.

We zoom further in– in Figure 4.5, we plot the ultra-low FPR region (top plot of the figure) and the rest of the FPR region (bottom plot of the figure). The legends of each plot record the partial AUC for each of the corresponding FPR sections, as well as the corresponding confidence intervals. In the ultra-low FPR regime ($\leq 0.01\%$), the original classifier (blue)

achieves the highest recall, exceeding 40%, while ROS and VAE classifiers drop to around 20% and 0%, respectively. However, while ultra-low FPRs may seem ideal, the recall levels in this region are often too low for deployment. Even the best classifier falls to the 40% recall range, which makes it unsuitable for production use. Instead, the 0.01%–0.1% range (bottom plot) yields better recall levels—up to and above 80%—while still maintaining tight FPR control. Within this region ($0.01\% \leq \text{FPR} \leq 0.1\%$), TVAE-augmented classifiers (green) consistently achieve the highest recall and partial AUC, outperforming ROS-augmented classifiers (orange) and baseline classifiers (blue). We also note that in this region ($0.01\% \leq \text{FPR} \leq 0.1\%$), ROS consistently outperforms the original baseline (reaching over 80% recall at 0.1% FPR), whereas before, one may have concluded ROS was a worse option than original baseline from simply looking at the entire ROC curve and total AUC.

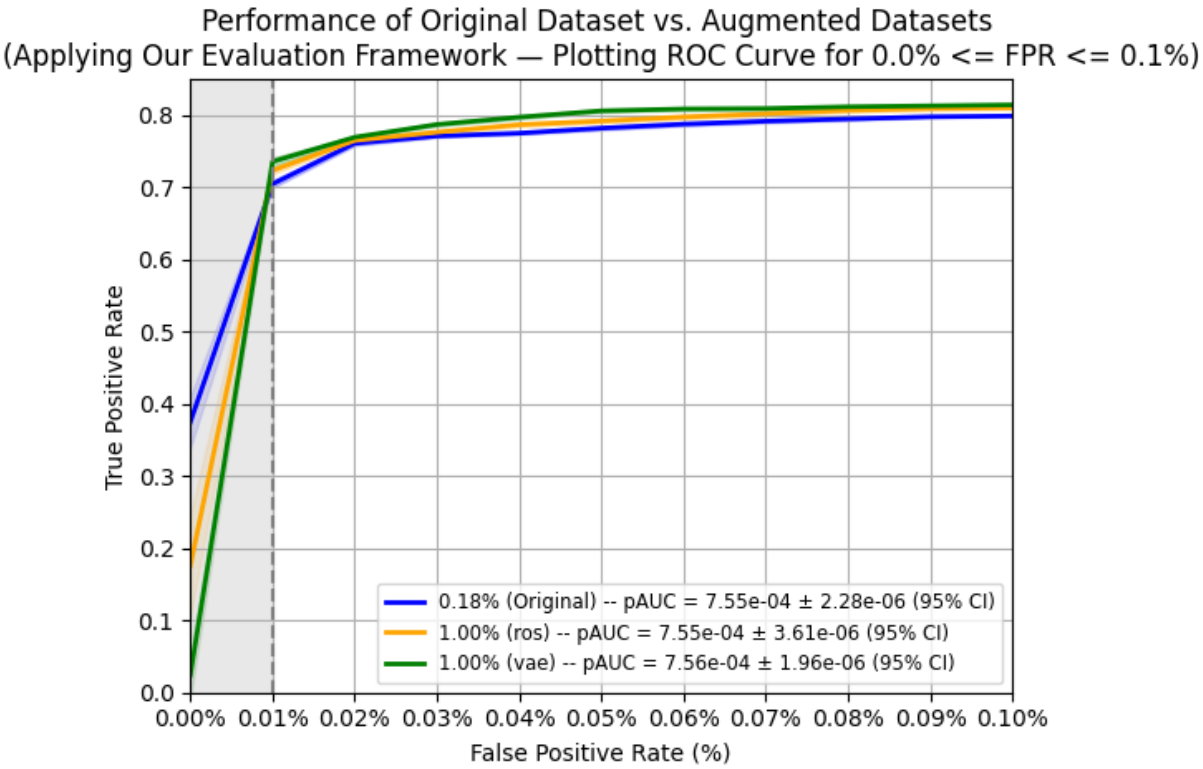

Figure 4.4: ROC curves under our evaluation framework, focused on the low-FPR region ($\leq 0.1\%$). The shaded gray area highlights the ultra-low FPR region ($\leq 0.01\%$).

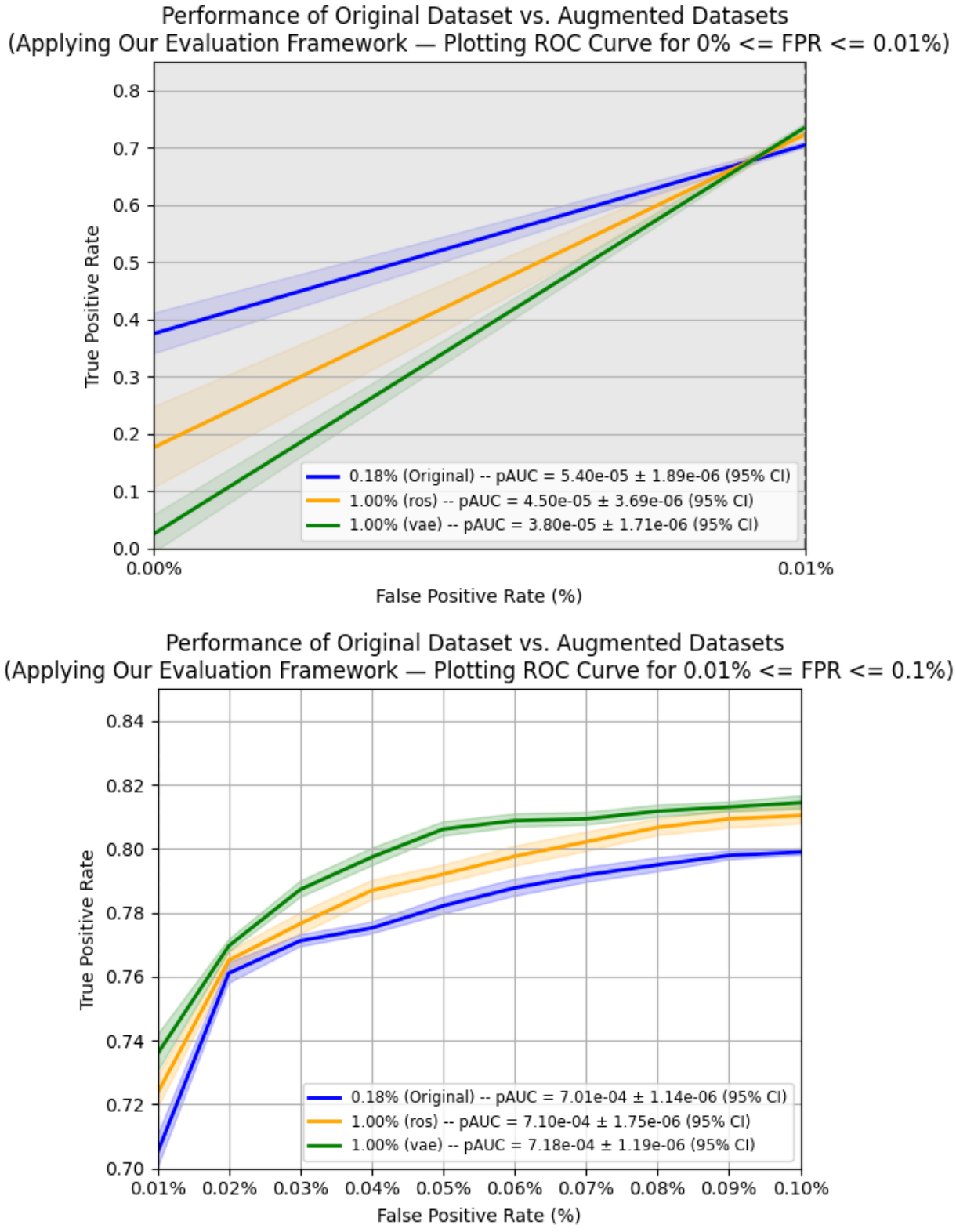

Figure 4.5: ROC curves under our Tail-end FPR Max Recall framework. Top: ultra-low FPR ($\leq 0.01\%$). Bottom: low FPR ($0.01\% \leq \text{FPR} \leq 0.1\%$).

Summary

In summary, to evaluate our classifiers, we began with the approach most academic papers follow: plotting the full ROC curve and computing the total area under the curve (AUC). From this, we observed that the TVAE-augmented classifiers consistently ranked higher across the FPR axis and achieved the highest total AUC, leading us to conclude that they outperformed the other dataset variants. We also noted that ROS-augmented classifiers generally fell below the baseline across most of the ROC curve and had the lowest total AUC—suggesting that Random Oversampling degraded performance in this setting.

However, we recognize that in real-world applications, businesses must impose an FPR budget—no matter how high the recall, a classifier allowing excessive false positives would be unusable in production. To simulate this, we set an operational FPR budget of 0.1%, meaning that only classifiers achieving FPR of 0.1% or lower would be considered viable. This shifted our focus to the leftmost portion of the ROC curve and required calculating the partial AUC within this constrained region.

Under this business-relevant constraint, we confirmed that TVAE-augmented classifiers not only maintained the highest partial AUC but also demonstrated consistently higher recall across the FPR-constrained region—validating our hypothesis that TVAE-based augmentation improves model performance in both academic and practical settings. Interestingly, this more targeted evaluation altered one of our earlier conclusions: under the 0.1% FPR budget, Random Oversampling actually outperformed the baseline in terms of recall, indicating that it may still be suitable for deployment under strict FPR constraints.

Ultimately, this analysis underscores the importance of evaluating classifiers within business-relevant FPR ranges. Our Tail-end FPR Max Recall framework addresses key gaps in academic evaluation practices by aligning model assessment with real-world deployment requirements.

4.3 Additional Work with Neural Embeddings

We also trained a neural network to generate feature embeddings from the tabular data, as described in Section 3.3.3. However, initial experiments did not yield consistent performance improvements when using the learned embeddings in place of or alongside the original features in the LightGBM model. As such, we do not report detailed metrics here, but note this as a direction for further tuning and evaluation.

Chapter 5

Discussion

5.1 Conclusion

This work presents a novel evaluation framework, Tail-end FPR Max Recall, that more accurately reflects the priorities of real-world institutions in anomaly detection systems. By concentrating on performance in the low false positive rate (FPR) regime, this framework addresses a critical operational constraint: minimizing false positives within a specific tolerance while maximizing high recall. This is essential in high-stakes environments like banking, where excessive false alarms can lead to customer dissatisfaction and increased operational costs.

Our experiments on a real credit card fraud dataset demonstrate that synthetic data generated via Tabular Variational Autoencoders (TVAE) leads to meaningful improvements in model performance. Classifiers trained on TVAE-augmented data outperform those trained on both the original dataset and the original dataset augmented using Random Oversampling. These improvements are consistently observed across both our proposed Tail-end FPR Max Recall metric and standard academic metrics, such as AUROC, highlighting the method's broad utility. Moreover, they underscore the potential of generative modeling—particularly TVAE—as a powerful tool for improving anomaly detection in highly imbalanced datasets.

By modeling the latent structure of the minority class, TVAE generates more varied and meaningful synthetic samples compared to duplication-based approaches like Random Oversampling.

Lastly, we conduct preliminary experiments using neural network–based feature embeddings to enhance the input representations for tree-based classifiers. Although these embeddings did not yield consistent performance improvements in our current setup, we see potential in further exploration. Techniques such as alternative architectures and user-specific embeddings may offer gains in future iterations.

5.2 Future Work

There are several potential directions for extending and building upon the contributions of this study. One direction is to apply the proposed Tail-end FPR Max Recall framework to additional classifier architectures, such as XGBoost [55], to assess whether the observed performance trends generalize across different modeling approaches. Expanding beyond LightGBM could help evaluate the broader applicability and robustness of the evaluation methodology.

Another promising direction involves scaling to larger and more representative datasets. While the MLG-ULB dataset reflects real transactional behavior, it is nowhere near the scale of the data that real-world financial institutions work in. The extreme class imbalance—fraud rates below 0.2% in both training and test sets—mirrors real-world conditions, but the small number of fraud cases also constrains the robustness of our findings. Larger-scale datasets would allow for more comprehensive evaluation, particularly in the low FPR and high-recall regimes.

Further exploration of synthetic data generation (SDG) methods may also yield useful insights. Our results suggest that TVAE improves classifier performance by learning a latent representation of the minority class and generating more diverse synthetic examples

compared to random oversampling. Future work could benchmark TVAE against other generative models—such as CTGAN or Gaussian Copulas—focusing on their utility for anomaly detection under realistic constraints.

Lastly, the use of deep feature embeddings remains an open area. Although our initial experiments did not yield consistent performance gains, future work could investigate alternative embedding strategies, such as Word2Vec-style encodings, user-specific representations, or broader architectural tuning. These approaches may offer more effective ways to integrate neural representations with tree-based models in the context of tabular anomaly detection.

References

[1] Y. Lucas and J. Jurgovsky. *Credit card fraud detection using machine learning: A survey*. 2020. arXiv: 2010.06479 [cs.LG]. URL: https://arxiv.org/abs/2010.06479.

[2] J. Morgan. *CNP Fraud Prevention: How to Combat Chargebacks*. Accessed: 2025-05-05. May 2024. URL: https://www.jpmorgan.com/insights/payments/analytics-and-insights/cnp-fraud-prevention-combat-chargebacks.

[3] F. Pro. *Anomaly Detection: Key Concepts and Applications*. Accessed: 2025-05-01. May 2024. URL: https://www.fb-pro.com/anomaly-detection/.

[4] C. P. S. Technologies. *Understanding False Negatives in Cybersecurity*. Accessed: 2025-05-01. May 2024. URL: https://www.checkpoint.com/cyber-hub/cyber-security/understanding-false-negatives-in-cybersecurity/#FalseNegativesvs.FalsePositives.

[5] Y. Cai, W. Zhang, H. Chen, and K.-T. Cheng. "MedIAnomaly: A comparative study of anomaly detection in medical images". In: *Medical Image Analysis* 102 (2025), p. 103500. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2025.103500. URL: https://www.sciencedirect.com/science/article/pii/S1361841525000489.

[6] T. Burt, K. S. Button, H. Thom, R. J. Noveck, and M. R. Munafò. "The Burden of the "False-Negatives" in Clinical Development: Analyses of Current and Alternative Scenarios and Corrective Measures". In: *Clinical and Translational Science* 10.6 (Nov. 2017). Epub 2017 Jul 4, pp. 470–479. DOI: 10.1111/cts.12478.

[7] H. Ma, A. I. Bandos, H. E. Rockette, and D. Gur. "On use of partial area under the ROC curve for evaluation of diagnostic performance". In: *Statistics in Medicine* 32.20 (Sept. 2013). Epub 2013 Mar 18, pp. 3449–3458. DOI: 10.1002/sim.5777.

[8] D. McElfresh, S. Khandagale, J. Valverde, V. P. C, B. Feuer, C. Hegde, G. Ramakrishnan, M. Goldblum, and C. White. *When Do Neural Nets Outperform Boosted Trees on Tabular Data?* 2024. arXiv: 2305.02997 [cs.LG]. URL: https://arxiv.org/abs/2305.02997.

[9] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed. "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms". In: *IEEE Access* 10 (2022), pp. 39700–39715. DOI: 10.1109/ACCESS.2022.3166891.

[10] S. Rezvani and X. Wang. "A Broad Review on Class Imbalance Learning Techniques". In: *Applied Soft Computing* 143 (2023), p. 110415. DOI: 10.1016/j.asoc.2023.110415.

[11] H. He and E. A. Garcia. "Learning from Imbalanced Data". In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), pp. 1263–1284.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-Sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357. DOI: 10.1613/jair.953.

[13] H. He, Y. Bai, E. A. Garcia, and S. Li. "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 2008, pp. 1322–1328. DOI: 10.1109/IJCNN.2008.4633969.

[14] M. Ahmed, A. N. Mahmood, and J. Hu. "A survey of network anomaly detection techniques". In: *Journal of Network and Computer Applications* 60 (2016), pp. 19–31. DOI: 10.1016/j.jnca.2015.11.016. URL: https://doi.org/10.1016/j.jnca.2015.11.016.

[15] H. He and E. 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. URL: https://doi.org/10.1109/TKDE.2008.239.

[16] Y. Shi et al. *lightgbm: Light Gradient Boosting Machine*. R package version 4.6.0.99. 2025. URL: https://github.com/Microsoft/LightGBM.

[17] P. Hajek, M. Z. Abedin, and U. Sivarajah. "Fraud Detection in Mobile Payment Systems using an XGBoost-based Framework". In: *Information Systems Frontiers* 2022 (2022). Online ahead of print, PMCID: PMC9560719, PMID: 36258679, pp. 1–19. DOI: 10.1007/s10796-022-10346-6.

[18] K. Ramani, I. Suneetha, N. Pushpalatha, and P. Harish. "Gradient Boosting Techniques for Credit Card Fraud Detection". In: *Journal of Algebraic Statistics* 13.3 (2022). Received 2022 April 02; Revised 2022 May 20; Accepted 2022 June 18, pp. 553–558. ISSN: 1309-3452. URL: https://publishoa.com.

[19] B. Xu, Y. Wang, X. Liao, and K. Wang. "Efficient fraud detection using deep boosting decision trees". In: *Decision Support Systems* 175 (2023), p. 114037. DOI: 10.1016/j.dss.2023.114037.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems* 30 (2017).

[21] K. Huang. "An Optimized LightGBM Model for Fraud Detection". In: *Journal of Physics: Conference Series* 1651.1 (Nov. 2020), p. 012111. DOI: 10.1088/1742-6596/1651/1/012111. URL: https://dx.doi.org/10.1088/1742-6596/1651/1/012111.

[22] IEEE-CIS and Vesta Corporation. *IEEE-CIS Fraud Detection Dataset*. Accessed: January 24, 2025. 2019. URL: https://www.kaggle.com/c/ieee-fraud-detection/data.

[23] A. A. Taha and S. J. Malebary. "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine". In: *IEEE Access* 8 (2020), pp. 25579–25587. DOI: 10.1109/ACCESS.2020.2971354.

[24] A. Vishnu Vardhan, P. V. S. N. M. L. Muppiri, D. S. Pasupuleti, M. Battula, and V. T. S. P. Muppuri. "Anomaly detection in credit card transactions using autoencoders". In: *International Journal of Advanced Research in Computer and Communication Engineering* 13.3 (2024), pp. 128–134. DOI: 10.17148/IJARCCE.2024.13320.

[25] J. Lin, X. Guo, Y. Zhu, S. Mitchell, E. Altman, and J. Shun. "FraudGT: A Simple, Effective, and Efficient Graph Transformer for Financial Fraud Detection". In: *Proceedings of the 5th ACM International Conference on AI in Finance (ICAIF '24)*. Brooklyn, NY, USA: ACM, 2024, p. 9. URL: https://doi.org/10.1145/3677052.3698648.

[26] F. Z. E. Hlouli, J. Riffi, M. A. Mahraz, and A. Yahyaouy. "Credit Card Fraud Detection Based on Multilayer Perceptron and Extreme Learning Machine Architectures". In: *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*. 2020, p. 9204185. DOI: 10.1109/ISCV49265.2020.9204185.

[27] C. Iscan and F. P. Akbulut. "Fraud Detection using Recurrent Neural Networks for Digital Wallet Security". In: *2023 8th International Conference on Computer Science and Engineering (UBMK)*. Burdur, Turkiye, Sept. 2023. URL: https://ieeexplore.ieee.org/document/10286651.

[28] I. Benchaji, S. Douzi, B. E. Ouahidi, and J. Jaafari. "Enhanced credit card fraud detection based on attention mechanism and LSTM deep model". In: *Journal of Big Data* 8 (Dec. 2021). DOI: 10.1186/s40537-021-00553-1. URL: https://doi.org/10.1186/s40537-021-00553-1.

[29] L. Jiang. "Detecting Scams Using Large Language Models". In: *arXiv preprint* (Feb. 2024). arXiv: 2402.03147 [cs.CR]. URL: https://arxiv.org/abs/2402.03147.

[30] G. Singh, P. Singh, and M. Singh. "Advanced Real-Time Fraud Detection Using RAG-Based LLMs". In: *arXiv preprint arXiv:2501.15290* (2025). License: CC BY 4.0. URL: https://arxiv.org/abs/2501.15290.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017. URL: https://arxiv.org/abs/1706.03762.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *Proceedings of NAACL-HLT*. 2018, pp. 4171–4186. URL: https://doi.org/10.18653/v1/N19-1423.

[33] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shinn, et al. "Language Models are Few-Shot Learners". In: *Proceedings of NeurIPS* 33 (2020), pp. 1877–1901. DOI: 10.48550/arXiv.2005.14165. URL: https://doi.org/10.48550/arXiv.2005.14165.

[34] L. Grinsztajn, E. Oyallon, and G. Varoquaux. "Why do tree-based models still outperform deep learning on tabular data?" In: *arXiv preprint arXiv:2207.08815* (2022). URL: https://doi.org/10.48550/arXiv.2207.08815.

[35] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. "Modeling Tabular Data Using Conditional GAN". In: *NeurIPS* (2019). DOI: 10.48550/arXiv.1907.00503.

[36] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin. "TabTransformer: Tabular Data Modeling Using Contextual Embeddings". In: *arXiv preprint arXiv:2012.06678* (2020). URL: https://arxiv.org/abs/2012.06678.

[37] V. Borisov, T. Leemann, and G. Kasneci. "Deep Neural Networks and Tabular Data: A Survey". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022). URL: https://arxiv.org/abs/2110.01889.

[38] D. Dablain, A. Khetan, A. Ogier, M. Xia, and S. Vikram. "Improving Tabular Deep Learning using Feature Selection and Feature Embedding". In: *arXiv preprint arXiv:2201.01812* (2022). URL: https://arxiv.org/abs/2201.01812.

[39] L. E. Dodd and M. S. Pepe. "Partial AUC Estimation and Regression". In: *Biometrics* 59.4 (2003), pp. 614–623. DOI: 10.1111/1541-0420.00071.

[40] R. Zhu, Y. Guo, and J.-H. Xue. "Adjusting the Imbalance Ratio by the Dimensionality of Imbalanced Data". In: *Pattern Recogn. Lett.* 133 (2020), pp. 1–7.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0.

[42] H. Bourlard and Y. Kamp. "Auto-association by multilayer perceptrons and singular value decomposition". In: *Biological Cybernetics* 59.4-5 (1988), pp. 291–294. DOI: 10.1007/BF00332918.

[43] M. Sakurada and T. Yairi. "Anomaly detection using autoencoders with nonlinear dimensionality reduction". In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. ACM. 2014, pp. 4–11. DOI: 10.1145/2689746.2689747.

[44] R. Chalapathy and S. Chawla. "Deep Learning for Anomaly Detection: A Survey". In: *arXiv preprint arXiv:1901.03407* (2019). URL: https://arxiv.org/abs/1901.03407.

[45] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *International Conference on Learning Representations (ICLR)*. 2014.

[46] D. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *International Conference on Learning Representations (ICLR)*. 2014. URL: https://openreview.net/forum?id=ZfJslz6lF7.

[47] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations (ICLR)*. 2017. URL: https://arxiv.org/abs/1611.02731.

[48] E. Jang, S. Gu, and B. Poole. "Categorical Reparameterization with Gumbel-Softmax". In: *International Conference on Learning Representations (ICLR)*. 2017. URL: https://arxiv.org/abs/1611.01144.

[49] K. Ramachandran, K. Kayathwal, H. Wadhwa, and G. Dhama. "FraudAmmo: Large Scale Synthetic Transactional Dataset for Payment Fraud Detection". In: *Proceedings of the 2023 International Joint Conference on Conference Name*. 2023.

[50] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson. "PAYSIM: A Financial Mobile Money Simulator for Fraud Detection". In: *Proceedings of the 28th European Modeling and Simulation Symposium 2016 (EMSS 2016)*. Larnaca, Cyprus: European Modeling and Simulation Symposium, Sept. 2016.

[51] Machine Learning Group - ULB. *Credit Card Fraud Detection Dataset*. Accessed: January 24, 2025. 2013. URL: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

[52] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, pp. 2623–2631.

[53] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[54] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[55] T. Chen and C. Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *KDD*. 2016. DOI: 10.48550/arXiv.1603.02754.