ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

MARCOS BRUM FREIRE

UNSUPERVISED DEEP LEARNING TO SUPERVISED INTERPRETABILITY: A DUAL-STAGE APPROACH FOR FINANCIAL ANOMALY DETECTION

Porto Alegre
2024

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

UNSUPERVISED DEEP LEARNING TO SUPERVISED INTERPRETABILITY: A DUAL-STAGE APPROACH FOR FINANCIAL ANOMALY DETECTION

MARCOS BRUM FREIRE

Master Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Duncan Dubugras Alcoba Ruiz

Porto Alegre
2024

APRENDIZADO PROFUNDO NÃO SUPERVISIONADO PARA MODELAGEM SUPERVISIONADA INTERPRETÁVEL: UMA ABORDAGEM EM DUAS FASES PARA DETECÇÃO DE ANOMALIAS FINANCEIRAS

RESUMO

A sofisticação crescente das atividades de lavagem de dinheiro demanda abordagens que aliem detecção eficaz de anomalias com interpretabilidade. Para enfrentar este desafio, propusemos uma arquitetura dual integrando um Autoencoder Variacional Auto-Adversarial com blocos transformadores para detecção não supervisionada de anomalias, associado a uma Máquina de Explainable Boosting para classificação supervisionada. Essa abordagem endereça limitações fundamentais na detecção de fraudes financeiras, como a escassez de dados rotulados e o desequilíbrio extremo de classes. Em avaliações realizadas com dados proprietários de transações financeiras, o framework alcançou uma Área Sob a Curva ROC de 0,9508 e uma Área Sob a Curva Precisão-Revocação de 0,5417. Quando aplicado ao conjunto de dados público de fraude em cartões de crédito, o modelo obteve uma Área Sob a Curva ROC de 0,964, superando métodos estabelecidos na literatura como Deep Autoencoder (0,882) e Autoencoder com Clustering (0,961), mesmo sem utilizar dados rotulados durante o treinamento. O componente Máquina de Explainable Boosting viabilizou a identificação clara dos fatores determinantes nas classificações de risco, enquanto o Autoencoder Variacional Auto-Adversarial demonstrou eficácia na detecção de padrões anômalos em diferentes contextos financeiros. Os resultados evidenciam o potencial desta solução integrada, que alia capacidade avançada de detecção à transparência necessária para aplicações práticas no setor financeiro.

Palavras-Chave: Detecção de Anomalias Financeiras, Autoencoders Variacionais Auto-Adversariais, Máquinas de Impulso Explicáveis, Inteligência Artificial Explicável, Aprendizado Profundo.

UNSUPERVISED DEEP LEARNING TO SUPERVISED INTERPRETABILITY: A DUAL-STAGE APPROACH FOR FINANCIAL ANOMALY DETECTION

ABSTRACT

The increasing sophistication of money laundering activities demands approaches that unite effective anomaly detection with interpretability. To address this challenge, we propose a dual-stage architecture integrating a Self-Adversarial Variational Autoencoder with transformer blocks for unsupervised anomaly detection, paired with an Explainable Boosting Machine for supervised classification. This approach addresses fundamental limitations in financial fraud detection, such as the scarcity of labeled data and extreme class imbalance. In evaluations on proprietary financial transaction data, the framework achieved a Receiver Operating Characteristic Area Under the Curve of 0.9508 and a Precision-Recall Area Under the Curve of 0.5417. When applied to the public credit card fraud dataset, the model attained a ROC AUC of 0.964, outperforming established methods in the literature such as Deep Autoencoder (0.882) and Autoencoder with Clustering (0.961), despite not using labeled data during training. The Explainable Boosting Machine component enabled clear identification of factors driving risk classifications, while the Self-Adversarial Variational Autoencoder component proved effective in detecting anomalous patterns across different financial contexts. The results demonstrate the potential of this integrated solution, which combines advanced detection capabilities with the transparency necessary for practical applications in the financial sector.

Keywords: Financial Anomaly Detection, Self-Adversarial Variational Autoencoders, Explainable Boosting Machines, Deep Learning, Interpretable AI.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

AE – Autoencoder

AML – Anti-Money Laundering

SAVAE – Self-Adversarial Variational Autoencoder

SAVAE-EBM – Self-Adversarial Variational Autoencoder - Explainable Boosting Machine

SAVAE-SR – Self-Adversarial Variational Autoencoder with Spectral Residual

C-VAE – Chaotic Variational Autoencoder

EBM – Explainable Boosting Machine

VAE – Variational Autoencoder

MSCVAE – Multi-Scale Convolutional Variational Autoencoder

ROC – Receiver Operating Characteristic

AUC – Area Under the Curve

ROC AUC – Area Under the ROC Curve (often used directly)

PR – Precision-Recall

PR AUC – Area Under the PR Curve (often used directly)

CNN – Convolutional Neural Network

CNN-BILSTM – Convolutional Neural Network - Bidirectional Long Short-Term Memory

LSTM – Long Short-Term Memory

LSTM-AE – Long Short-Term Memory Autoencoder

1DCONV-LSTM – 1D Convolutional LSTM

BILSTM – Bidirectional Long Short-Term Memory

GAN – Generative Adversarial Network

WGAN – Wasserstein Generative Adversarial Network

LSTM-VAE-GAN – LSTM-based Variational Autoencoder - Generative Adversarial Network

TADGAN – Time series Anomaly Detection using GAN

BEATGAN – Bidirectional GAN

XAI – Explainable Artificial Intelligence

ML – Machine Learning

DRL – Deep Reinforcement Learning

GNN – Graph Neural Network

NENN – Node and Edge Neural Network

HONN – Higher-Order Neural Network

SVM – Support Vector Machine

LOF – Local Outlier Factor

RF – Random Forest

DBDT – Deep Boosting Decision Tree

GAM – Generalized Additive Model

CIU – Contextual Importance and Utility

MAE – Mean Absolute Error

MSE – Mean Squared Error

MAPE – Mean Absolute Percentage Error

MCC – Matthews Correlation Coefficient

KPI – Key Performance Indicator

NYSE – New York Stock Exchange

PCA – Principal Component Analysis

VAR – Value-at-Risk

ARIMA – Autoregressive Integrated Moving Average

CONTENTS

1. INTRODUCTION

Money laundering represents a critical mechanism that enables criminals to inject illicitly obtained proceeds[1] into the legitimate financial system. The International Monetary Fund estimates that between 2–3% of the total world gross domestic product undergoes laundering annually [36]. This criminal activity has evolved significantly, with formal criminalization occurring relatively recently; for instance, the United States first declared it a crime in 1986, while some jurisdictions like Saudi Arabia implemented comprehensive criminalization as late as 2003 [36]. Furthermore, money laundering exhibits strong systemic connections with various criminal enterprises, functioning as both a mechanism that enables predicate criminal acts and a source for subsequent criminal activities [36]. Research indicates that approximately 15-20% of laundered money is reinvested in financing new criminal activities, with this proportion showing an upward trend of about 50% over recent years [36].

However, financial institutions face significant challenges in their anti-money laundering (AML) efforts.Current rule-based systems demonstrate severe limitations, with over 90% of alerts being false positives [22], which creates substantial operational burdens. Specifically, the current transaction monitoring infrastructure suffers from three primary weaknesses. Firstly, the high volume of false positives requires extensive human resources for review, leading to inconsistent performance and potential oversight of genuine suspicious activities [22]. Secondly, criminals can circumvent detection by exploiting knowledge of rule-based systems through various channels, including insider threats and published typologies [22]. Lastly, the development and implementation of new rules against emerging money laundering methods remains a lengthy and reactive process [35].

In this context, applying machine learning to AML presents various challenges. While supervised learning methods have demonstrated effectiveness in specific contexts, such as Bitcoin transaction classification [2], their broader application often faces limitations due to the scarcity of labeled data and the highly imbalanced nature of financial fraud datasets [25]. Deep learning methods like CNN variants, AutoEncoder, and graph-based neural networks have been explored for identifying complex money laundering patterns [25]. However, the financial sector’s need for interpretability and transparency in model decisions requires integrating techniques like Explainable Artificial Intelligence (XAI) [25].

This thesis proposes a novel dual-stage approach to anomaly detection in financial transactions, explicitly targeting potential fraudulent activities. The methodology combines a Self-Adversarial Variational Autoencoder (SAVAE) enhanced with transformer blocks for unsupervised anomaly detection with an Explainable Boosting Machine (EBM)

[1] According to the Financial Action Task Force (FATF), *Proceeds* refers to "any property derived from or obtained, directly or indirectly, through the commission of an offence" [15].

for supervised classification and interpretability. Our framework aims to address key challenges in the field:

- The unsupervised nature of the SAVAE component enables effective pattern learning from unlabeled financial transaction data.
- The incorporation of transformer blocks enhances the model's capacity to capture complex temporal dependencies inherent in financial transaction sequences.
- The integration of the EBM provides crucial interpretability of the model's classification decisions, facilitating trust and understanding.
- The dual-stage architecture facilitates both the identification of anomalous transactions and the precise classification of risk levels.

We demonstrate the effectiveness of this approach through comprehensive experimentation on both a proprietary financial transaction dataset and a publicly available credit card fraud dataset. The results indicate strong performance. The model achieved a Receiver Operating Characteristic Area Under the Curve (ROC AUC) of 0.9508 and a Precision-Recall Area Under the Curve (PR AUC) of 0.5417 on the proprietary dataset, with comparable performance on the public dataset. The framework offers interpretable insights while ensuring high detection accuracy, making it a valuable tool for strengthening AML efforts.

Building on these outcomes, this thesis demonstrates the effective integration of unsupervised deep learning with supervised models to enhance financial anomaly detection. The unsupervised model provides a strong anomaly detector, while the Explainable Boosting Machine (EBM) adds transparency, making the approach well-suited for both fraud detection and reporting.

This thesis is organized as follows: Chapter 2 provides essential background information on supervised learning, unsupervised learning, and deep learning concepts relevant to this work. Chapter 3 presents related work in money laundering and anomaly detection, focusing on recent advances in self-adversarial approaches and interpretable machine learning models. Chapter 4 outlines the research objectives and research questions. Chapter 5 presents the methodology and evaluation metrics, including a detailed description of the proposed SAVAE-EBM framework. Chapter 6 details the experimental results and provides a thorough analysis. Finally, Chapter 7 concludes the thesis, addressing the framework's limitations and proposing future work in interpretability, semi-supervised learning, and domain knowledge integration.

2. BACKGROUND

2.1 Machine Learning Applications in AML

Recent advances in machine learning offer promising solutions for enhancing AML effectiveness. Studies demonstrate that deep learning techniques can significantly decrease false positives compared to traditional classifiers [35]. Current research indicates specific performance improvements.

2.1.1 Performance Metrics

Empirical studies show that machine learning models using time-frequency features alone can achieve a false positive rate of 14.9% with an F-score of 59.05%. When these features are combined with traditional transaction and customer relationship management data, performance improves further to an 11.85% false positive rate and a 74.06% F-score [22].

2.2 Client Risk Classification

Modern approaches increasingly focus on client risk classification as a fundamental component of detection strategies. Expert surveys indicate varying rates of detection across different types of criminal proceeds:

> "Proceeds from economic and financial crimes are laundered in 55–60% of cases; proceeds from drug trafficking are laundered in 45–50% of cases; proceeds from illegal arms trafficking are laundered in 35–40% of cases" [36, p. 866].

The above figures are taken directly from the survey results presented by Rusanov and Pudovochkin, highlighting the complexity and prevalence of money laundering across different crime types.

2.3 Variational Autoencoders

Variational Autoencoders (VAEs), introduced by Kingma and Welling [23], address the challenge of efficient inference and learning in directed probabilistic models with continuous latent variables and intractable posterior distributions. They offer a framework for

performing efficient approximate inference and learning with continuous latent variables [23]. The VAE architecture consists of an encoder network $q_\phi(z|x)$, also known as a recognition model, which approximates the intractable true posterior distribution $p_\theta(z|x)$. The decoder network $p_\theta(x|z)$, referred to as the generative model, works together with a prior distribution $p_\theta(z)$ to form the complete generative model [23].

A key innovation of VAEs is the reparameterization trick, which enables gradient-based optimization through the stochastic sampling process [23]. For a Gaussian approximate posterior, this involves reparameterizing a sample z as $z = \mu + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, enabling backpropagation. The model parameters are optimized by maximizing a variational lower bound (ELBO) on the marginal likelihood [23]:

$$\mathcal{L}(\theta, \phi; x) = -D_{KL}(q_\phi(z|x)||p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$$

This objective function comprises two components. The KL-divergence term serves as a regularizer, guiding the approximate posterior to align closely with the prior. The expected reconstruction error term ensures accurate data modeling [23]. Through extensive experimentation, the authors demonstrate that VAEs can efficiently handle large datasets and provide both theoretical advantages and strong experimental results [23]. The method proves particularly effective for continuous latent variables and scales well to large datasets, offering a practical approach to learning complex probability distributions.

2.4 Self-Adversarial Variational Autoencoder

The Self-Adversarial Variational Autoencoder (SaVAE) [50, 28] extends the Variational Autoencoder (VAE) framework for improved anomaly detection. Unlike standard VAEs, SaVAEs incorporate adversarial objectives to enhance discriminative power and prevent overfitting to normal data.

SaVAEs employ two key mechanisms:

1. **Adversarial Encoder-Generator Interaction:** The encoder acts as both a probabilistic mapper to latent space and a discriminator between real and reconstructed data, while the generator aims to produce realistic outputs to deceive the encoder.
2. **Gaussian Anomaly Priors:** The model assumes Gaussian distributions for both normal and anomalous data in the latent space, using a Gaussian transformer network to synthesize anomalous latent variables, enabling the generator to distinguish between normal and anomalous regions.

This adversarial VAE framework is designed to offer several advantages over conventional VAEs, such as regularization of the generator, explicit anomaly detection via

synthesized anomalous latent variables, and a more compact model structure. Whether these advantages hold in practice remains to be investigated.

2.4.1 Self-Adversarial Variational Autoencoder Architecture

The SaVAE architecture comprises an encoder (E), a generator (G), and a Gaussian transformer (T). Training alternates between updating (G and T) with fixed E, and updating E with fixed (G and T) [50]. Figure 2.1 illustrates the architecture and training flow of the SaVAE model.

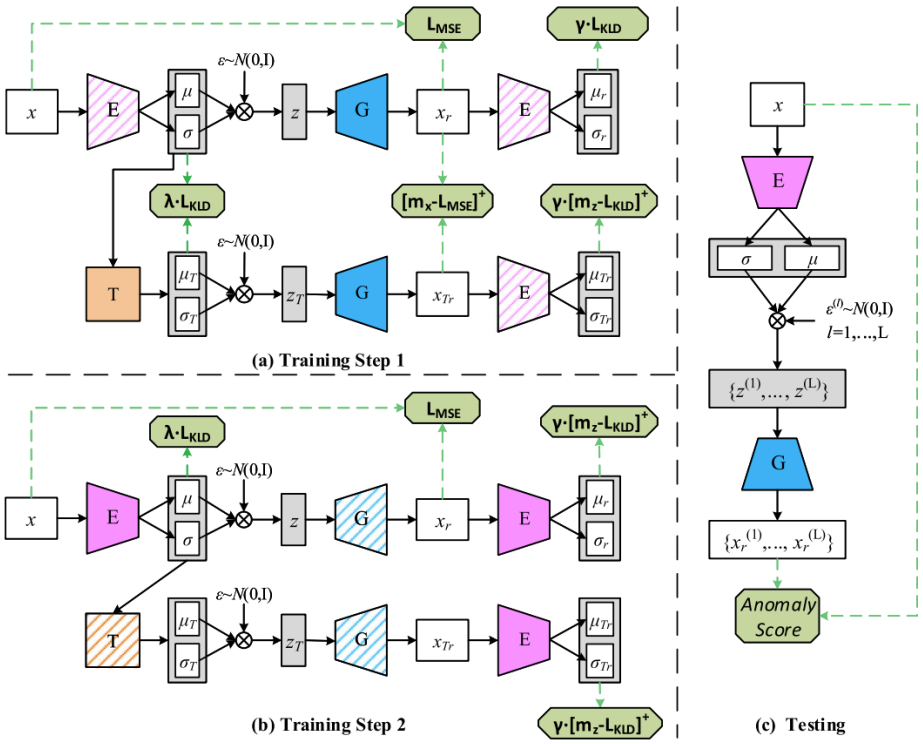

Figure 2.1 – SaVAE architecture and training flow. Reproduced from Wang et al. [50].

2.5 The Transformer Architecture

The emergence of the Transformer architecture marked a significant shift from traditional sequence transduction models. Unlike its predecessors that relied on recurrent or convolutional neural networks, the Transformer introduces a novel approach based entirely on attention mechanisms [48] This architectural innovation has proven particularly effective in handling sequential data while offering enhanced parallelization capabilities.

2.5.1 Core Architecture

The Transformer uses an encoder-decoder structure, where both components implement stacked self-attention and point-wise fully connected layers. The encoder consists of six identical layers, each featuring a multi-head self-attention mechanism along with a position-wise fully connected feed-forward network. Likewise, the decoder consists of six identical layers and features an additional third sub-layer for multi-head attention applied to the output of the encoder.Figure 2.2 illustrates this architecture.

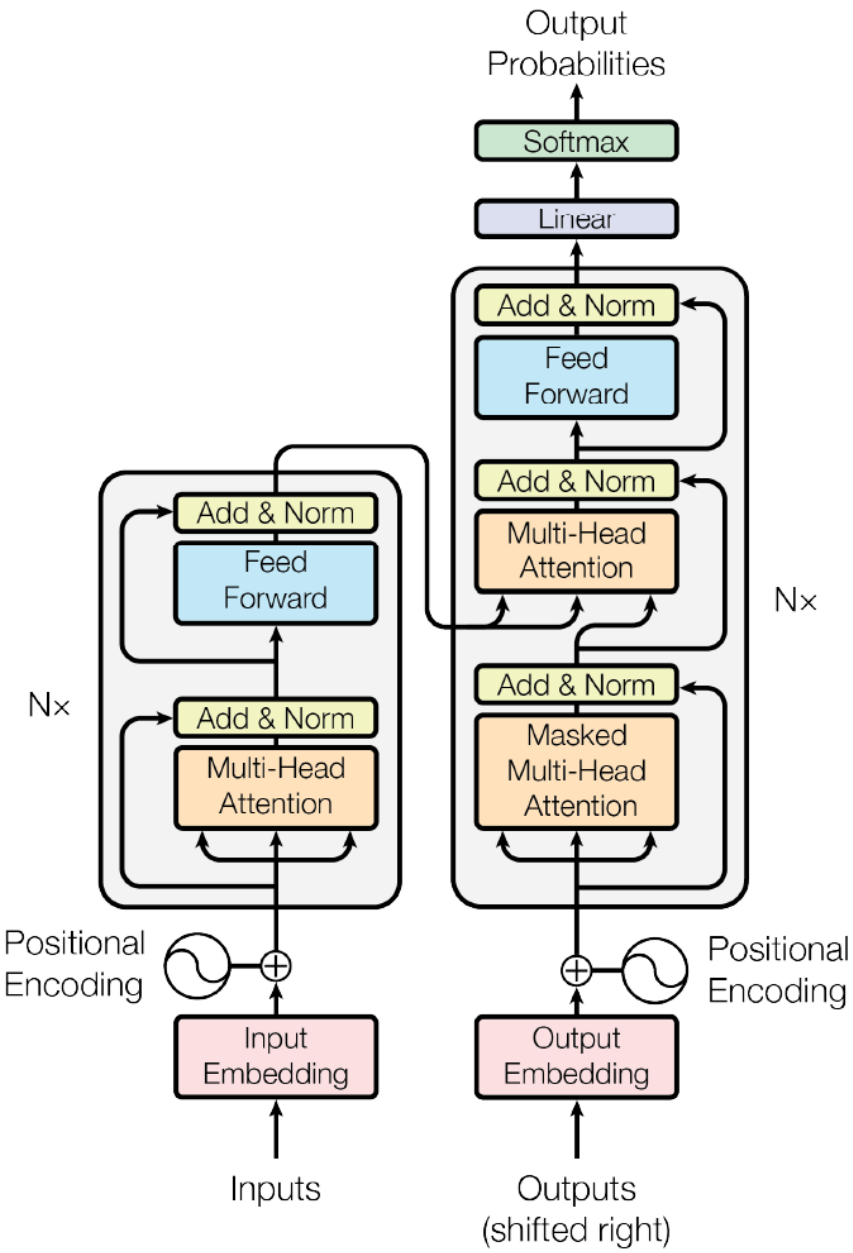

Figure 2.2 – The Transformer architecture, consisting of encoder and decoder stacks. The encoder maps the input sequence to continuous representations, while the decoder generates the output sequence. Reproduced from [48].

2.5.2 Attention Mechanism

At the heart of the Transformer lies its attention mechanism, which computes the relationship between queries and key-value pairs. The model employs what the authors term "Scaled Dot-Product Attention," calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

where Q represents queries, K represents keys of dimension d_k, and V represents values. The scaling factor $\frac{1}{\sqrt{d_k}}$ prevents the dot products from growing too large in magnitude, which could push the softmax function into regions with extremely small gradients [48].

2.5.3 Multi-Head Attention

Rather than performing a single attention function, the Transformer implements multi-head attention, allowing it to attend to information from different representation subspaces simultaneously. This approach projects queries, keys, and values h times with different learned linear projections. Each projection creates attention heads that operate in parallel, with their outputs concatenated and linearly transformed to produce the final values [48]. The multi-head attention mechanism is also shown in Figure 2.3.

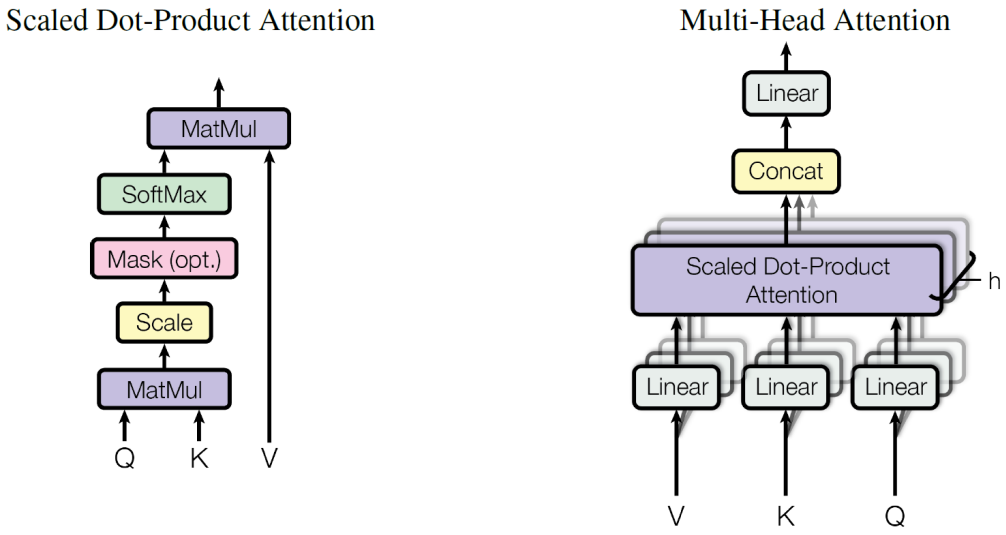

Figure 2.3 – (left) Scaled Dot-Product Attention mechanism. (right) Multi-Head Attention consisting of several attention layers operating in parallel. Reproduced from [48].

2.5.4 Positional Encoding

Since the Transformer contains no recurrence or convolution, it incorporates positional encodings to leverage sequence order. These encodings utilize sine and cosine functions at various frequencies:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \tag{2.2}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \tag{2.3}$$

This approach enables the model to attend to relative positions, as for any fixed offset k, PE_{pos+k} can be represented as a linear function of PE_{pos} [48].

2.5.5 Advantages Over Traditional Architectures

The Transformer architecture offers several key advantages over recurrent neural networks. It reduces the number of sequential operations required to relate signals from different positions in the input sequence to a constant number, whereas RNNs require O(n) sequential operations. This characteristic enables better parallelization and more efficient training, particularly for longer sequences [48].

2.6 Performance Metrics for Imbalanced Datasets

The evaluation of binary classification models, particularly in scenarios with imbalanced class distributions, requires careful consideration of appropriate performance metrics. Traditional metrics may provide misleading interpretations when dealing with rare events or minority classes [12].

2.6.1 Fundamental Metrics

We define several basic metrics in binary classification using the confusion matrix elements. Table 2.1 summarizes the fundamental metrics derived from these elements.

Table 2.1 – Evaluation Metrics for Binary Classification

Metric	**Formula**
Precision	$\frac{TP}{TP+FP}$
Recall (Sensitivity)	$\frac{TP}{TP+FN}$
False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
Specificity	$\frac{TN}{TN+FP}$
Accuracy	$\frac{TP+TN}{TP+FN+TN+FP}$
F1-Score	$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$

2.6.2 ROC and PR Curves

Receiver Operating Characteristic (ROC) curves and Precision-Recall (PR) curves represent two primary approaches to evaluating binary classifiers. While ROC curves plot recall against the false-positive rate, PR curves plot precision against recall. The area under these curves (AUC-ROC and AUC-PR) provides aggregate measures of performance [40].

ROC curves maintain invariance to class distribution, making them stable across different class ratios. However, this property can mask important performance variations in highly imbalanced scenarios [12]. PR curves, conversely, show higher sensitivity to imbalanced distributions, making them more suitable for evaluating rare event detection [40].

2.6.3 Metric Selection Considerations

The choice between ROC and PR-based metrics depends significantly on the application context. PR curves prove more informative in scenarios with rare events or highly imbalanced datasets, as they give greater emphasis to false positives [12]. This makes them particularly suitable for applications such as fraud detection or disease diagnosis, where the positive class represents a small minority of cases.

3. RELATED WORK

3.1 Variational Autoencoders in Anomaly Detection

The effectiveness of VAEs in financial anomaly detection has been demonstrated in various contexts. For example, Tingfei Huang *et al.* [44] showed that VAEs outperform traditional methods like Support Vector Machines, Logistic Regression, and k-Nearest Neighbors, as well as other deep learning approaches, in credit card fraud detection. They used a VAE-based oversampling technique to address class imbalance in credit card fraud datasets. Their approach, evaluated on a publicly available credit card fraud dataset [46], yielded significant improvements in precision and F1-score compared to SMOTE and GAN-based oversampling methods. Specifically, by injecting synthetic fraudulent transactions equivalent to 50% of the original number into the training set, their VAE method achieved a precision of approximately 0.90 and an F1-score of approximately 0.87. These results highlight the capacity of VAEs to learn robust representations that effectively distinguish legitimate from fraudulent transactions, leveraging their probabilistic nature to capture uncertainty and improve decision reliability.

3.2 Key Studies Implementing VAEs

Recent advancements in Variational Autoencoder architectures have significantly enhanced time series anomaly detection, particularly for multivariate time series data common in financial applications. These advancements focus on addressing challenges like multi-scale temporal patterns and data imbalance, which are crucial for effectively identifying anomalies in complex financial transactions.

Yokkampon *et al.* [55] introduced the Multi-Scale Convolutional Variational Autoencoder (MSCVAE) to effectively detect anomalies across multiple temporal scales. This approach leverages convolutional layers at different scales to capture both short-term and long-term dependencies within the time series data. Their evaluation on four benchmark datasets (Satellite, Wafer, EEG, and Opt) demonstrated substantial improvements over baseline methods, including standard VAEs and LSTM-AEs. For example, on the Satellite dataset, MSCVAE achieved a precision of 0.9683, recall of 0.9531, and F1-score of 0.9606, significantly outperforming the LSTM-AE (F1: 0.7216) and standard VAE (F1: 0.8400). Importantly, MSCVAE demonstrated robustness across varying anomaly rates (1% to 20%), highlighting its ability to handle imbalanced datasets, a common characteristic of AML data.

Niu *et al.* [31] proposed a different approach by combining the temporal modeling strengths of Long Short-Term Memory (LSTM) networks with the generative capabilities of a Generative Adversarial Network (GAN) in their LSTM-based VAE-GAN model. This architecture effectively captures temporal dependencies in time series data while maintaining computational efficiency through the integration of encoder mapping and discriminative capabilities. Tested on the Yahoo A1Benchmark and KPI datasets, the model achieved F1-scores of 0.8907 and 0.6552, respectively. Notably, the LSTM-VAE-GAN demonstrated a significant reduction in detection time compared to conventional GAN-based anomaly detection methods, making it suitable for real-time applications where timely anomaly detection is crucial.

3.2.1 Self-Adversarial Variational Autoencoder with Spectral Residual (SaVAE-SR)

The SaVAE-SR model, developed by Liu *et al.* [28], introduces a novel unsupervised approach specifically designed for time series anomaly detection. By integrating spectral residual (SR) techniques, SaVAE-SR preprocesses raw time series data into saliency maps, which act as pseudo-labels highlighting anomalous regions. This preprocessing step reduces the reliance on manually labeled data, a significant advantage in AML where labeled data is often scarce. Evaluated on datasets exhibiting characteristics similar to financial transactions, such as the KPI and Yahoo S5 datasets, SaVAE-SR demonstrated substantial performance improvements. The use of SR significantly reduced false positive rates (by up to 25% compared to conventional VAEs) while maintaining high recall rates (achieving 100% recall on the KPI dataset). These findings highlight the effectiveness of SR in mitigating anomaly contamination and addressing label scarcity, crucial aspects of AML datasets.

Compared to baseline methods, SaVAE-SR outperformed Donut, LSTM-based methods, and traditional statistical models by achieving F1-scores exceeding 0.9 on the Yahoo S5 dataset. Its ability to balance detection accuracy and computational efficiency makes it particularly suitable for applications in anomaly detection tasks like AML. While its direct application to AML datasets is yet to be fully established, the strong performance on imbalanced and complex data highlights its potential for financial applications.

3.2.2 Chaotic Variational Autoencoders (C-VAEs)

Building upon the VAE architecture, Gangadhar *et al.* [17] introduced Chaotic-VAE (C-VAE), which leverages chaotic maps within the latent space to enhance the detection of subtle anomalies. By incorporating logistic chaotic maps, C-VAE captures the non-linear

and stochastic dynamics often present in financial time series, improving its ability to discern subtle deviations indicative of fraudulent behavior. Evaluated on Medicare and automobile insurance datasets, C-VAE demonstrated a 5–10 percentage point improvement in classification rates compared to baseline VAEs, achieving 77.9% and 87.25% accuracy, respectively. The chaotic mapping approach proved especially effective in scenarios with extreme class imbalance, a characteristic often observed in fraud detection and AML, where fraudulent instances are significantly less frequent than legitimate transactions. This suggests C-VAE's potential for improving anomaly detection in AML by capturing intricate patterns often missed by traditional methods.

3.2.3 Synthetic Data Augmentation with WGANs

The lack of labeled anomaly data presents a significant challenge in training effective AML models. To address this, Chen *et al.* [11] proposed using Wasserstein Generative Adversarial Networks (WGANs) to generate synthetic fraudulent transactions. By augmenting training datasets with these synthetic samples, WGANs help mitigate class imbalance and improve the model's sensitivity to rare events. When combined with AE/VAE models, this data augmentation strategy has demonstrated improvements in the detection of fraudulent activities, addressing the issue of class imbalance. This approach is particularly relevant to AML, where the ratio of normal to fraudulent transactions can be extremely high (often exceeding 20,000:1). By generating realistic synthetic fraudulent transactions, Wasserstein GANs (WGANs) enhance the training of more effective anti-money laundering (AML) detection models, addressing the challenge of learning from limited labeled data.

3.2.4 Implications for Financial Applications

The combination of adversarial training, spectral residual analysis, and synthetic data augmentation positions SAVAEs and their derivatives as a leading framework for anomaly detection in financial applications, including AML. SaVAE-SR's demonstrated ability to handle imbalanced datasets, coupled with C-VAE's efficacy in capturing non-linear patterns and the benefits of WGAN-based data augmentation, highlights their potential for detecting complex money laundering schemes. While further research is needed to evaluate their direct performance on AML-specific datasets, the existing evidence suggests that these advanced VAE architectures can significantly contribute to more robust and effective AML detection systems.

3.3 Deep Learning Approaches in Anomaly Detection for Financial Markets

The dynamic nature of financial markets creates unique challenges for anomaly detection, requiring models that can capture intricate temporal patterns and adapt to changing market conditions. Deep learning is a promising approach for tackling complex challenges by learning intricate representations from data. This section examines various deep learning architectures and hybrid approaches applied to financial market anomaly detection, highlighting their strengths, limitations, and relevance to AML.

3.3.1 Time Series Analysis

Recurrent architectures, particularly Long Short-Term Memory (LSTM) networks, have been widely applied to analyzing financial time series due to their ability to capture temporal dependencies. However, traditional LSTMs suffer from limitations such as the vanishing gradient problem, where error signals diminish rapidly during backpropagation, especially over long sequences [27], and difficulties capturing very long-range dependencies, which can hinder their performance in complex financial markets. Recent work has explored more sophisticated combinations and modifications to address these limitations.

Naidoo and Du [30] developed a hybrid model combining LSTM autoencoders with Higher-Order Neural Networks (HONNs) to address stock market unpredictability. Using historical NYSE data from Yahoo Finance with labeled outlier events derived from company reports and news articles, their approach achieved a mean absolute percentage error (MAPE) of 0.03% and a validation loss of 0.0021, demonstrating superior performance compared to traditional LSTM and GRU models. This demonstrates the potential of hybrid architectures to improve predictive accuracy in noisy financial data.

Similarly, Yang *et al.* [54] proposed a 1D Convolutional LSTM (1dConv-LSTM) architecture for the Chinese stock market. By analyzing daily stock price data from 2015 to 2019 for 13 stocks, their model achieved mean absolute error (MAE) values consistently below 4.0, though the specific values varied across different stocks. They also reported an improved mean squared error (MSE) of 0.0171 on validation datasets. Their work highlights the ability of deep learning to identify anomalous price fluctuations. The application of transformers to financial anomaly detection has also gained traction. An et al. [3] introduced Finsformer, a transformer-based model with a novel cluster-attention mechanism. Evaluated against RNN, LSTM, and BERT models on a financial transaction dataset, Finsformer achieved impressive precision, recall, and accuracy scores of 0.97, 0.94, and 0.95, respectively. Their ablation studies further confirmed the effectiveness of the Transformer

architecture in processing complex financial data, demonstrating its potential for capturing intricate patterns and relationships indicative of anomalous behavior.

Crépey *et al.* [13] introduced an innovative integration of Principal Component Analysis (PCA) with Neural Networks for financial time series analysis. Their method, tested on both synthetic geometric Brownian motion data and real-world credit card fraud cases, achieved F1 scores up to 94.38% on synthetic datasets. The integration improved value-at-risk (VaR) estimation accuracy after anomaly removal, suggesting its potential for risk management in financial institutions.

3.3.2 Deep Reinforcement Learning

Deep reinforcement learning (DRL) offers a distinct approach to anomaly detection by training agents to learn optimal strategies in dynamic environments. Arshad *et al.* [4] conducted a comprehensive review of DRL applications in financial markets, highlighting its potential for anomaly detection. Their analysis suggests that DRL models can outperform traditional supervised and unsupervised methods, especially for large-scale unlabeled datasets. This makes DRL particularly attractive for AML, where labeled data is often scarce. However, implementing and fine-tuning DRL models can be challenging due to their complexity and the need for carefully designed reward functions.

Tallboys *et al.* [42] explored unsupervised techniques, including LSTM-based approaches with dynamic thresholding, for stock market manipulation detection. Using five labeled real-world datasets related to stock market manipulations, they found that while deep learning approaches identified anomalous areas effectively, traditional methods like ARIMA offered faster and often more precise identification of anomalous periods. This suggests that while deep learning has potential, simpler models may still be effective in certain contexts.

3.3.3 Hybrid and Graph-Based Approaches

Recent research increasingly focuses on hybrid approaches that combine deep learning with other techniques. Reddy *et al.* [34] demonstrated the efficacy of combining Convolutional Neural Networks (CNNs) with Bidirectional LSTMs for financial fraud detection. Their CNN-BiLSTM model, enhanced with Kernel PCA for feature extraction, achieved high accuracy (97.54%) in identifying fraudulent transactions, significantly outperforming standalone CNN and BiLSTM models.

Graph-based approaches are particularly relevant for AML due to their ability to model relationships between entities. In financial contexts, graph data structures can

represent transactions as edges connecting nodes that represent accounts or individuals. This allows for the detection of suspicious patterns within transaction networks, such as unusual flows of money or connections to known illicit actors. Silva *et al.* [41] highlighted the potential of Graph Neural Networks (GNNs), specifically Node and Edge Neural Networks (NENN), for detecting money laundering. Using a synthetic transaction dataset generated by AMLSim [51], they showed that representing transactions as edges performs better with higher class imbalance, achieving an F1-score of 74.51% for illicit transactions. AMLSim is a multi-agent simulation platform that generates synthetic transaction data by first creating a graph of accounts based on a degree distribution and then simulating transactions on this graph based on real-world transaction patterns. This allows researchers to experiment with different graph structures and transaction dynamics in a controlled environment. This work on NENNs complements that of Assumpção *et al.* [5], who developed DELATOR, a multi-task learning framework based on GNNs that leverages link prediction and edge regression for improved money laundering detection on large transaction graphs. Their approach, which also leverages synthetic data from AMLSim, led to a 23% improvement in AUC-ROC score compared to existing solutions, demonstrating the value of GNN-based methods for AML.

Dileep *et al.* [14] combined deep neural networks with Random Forest (RF) and Support Vector Machines (SVM) for financial fraud detection. Using the FraudTrain and FraudTest datasets from Kaggle, they achieved the highest accuracy of 99% with the Local Outlier Factor (LOF) algorithm, maintaining a recall of 0.8. Mizher and Nassif [29] addressed the challenge of skewed datasets in credit card fraud through a hybrid approach combining Deep Convolutional Neural Networks with SVM and RF techniques. Testing on a real-world imbalanced credit card fraud detection dataset, they achieved an impressive accuracy of 99.7% with the Random Forest classifier, outperforming other comparative models.

3.3.4 Relevance to AML

These diverse deep learning approaches offer valuable insights for developing robust AML systems. The ability of time series models like LSTM and hybrid architectures to capture temporal dependencies is crucial for analyzing transaction sequences. Similarly, the potential of DRL to learn optimal detection strategies in dynamic environments, and the capacity of GNNs to leverage relational information, aligns well with the complexities of AML. While these methods have primarily been applied to related financial domains, their demonstrated ability to handle imbalanced data, complex patterns, and noisy environments makes them promising candidates for future research in AML.

3.4 Integration of Transformers with VAEs

Transformers, renowned for their ability to capture long-range dependencies in sequential data through self-attention mechanisms, have recently been integrated with VAEs to enhance anomaly detection capabilities. This integration leverages the strengths of both architectures: the probabilistic framework of VAEs for learning data distributions and the powerful feature extraction capabilities of transformers.

A key advantage of transformers over recurrent networks like LSTMs is their parallel processing sequences, leading to significantly faster training times [48]. Recurrent models inherently rely on sequential computations, which limits parallelism and slows down training, especially with longer sequences [48]. The Transformer, by utilizing self-attention mechanisms instead of recurrence, allows for significantly more parallelization, which is especially beneficial for handling large financial datasets that require timely updates [45]. Furthermore, the self-attention mechanism in the Transformer can capture dependencies between input positions regardless of their distance, enabling it to model long-range dependencies, unlike recurrent models [48].

The AnoFormer model [39] exemplifies the effectiveness of integrating transformers within a GAN framework for anomaly detection. Its two-step masking strategy, involving random masking followed by entropy-based re-masking, enhances the model's ability to learn robust representations of normal data and identify deviations. This approach has shown superior performance compared to CNN- and LSTM-based VAEs on benchmark datasets. By replacing the convolutional or recurrent components of traditional VAEs with transformer blocks, the model can capture global temporal trends more effectively while maintaining computational efficiency.

Integrating transformers with VAEs specifically addresses some of the challenges in AML. The ability to process large transaction datasets efficiently and capture long-range dependencies is crucial for identifying complex money laundering patterns that may involve transactions spread over time and across multiple accounts. The improved feature extraction capabilities of transformers can enhance the model's sensitivity to subtle anomalies, while the probabilistic framework of VAEs provides a measure of uncertainty quantification, which is valuable for risk assessment in AML. This combination offers a promising avenue for developing more robust and interpretable AML detection systems.

3.5 Transformer-Based Anomaly Detection Models

Transformer architectures have revolutionized sequence modeling through their innovative attention mechanism [48]. This architectural advancement holds significant

promise for anomaly detection, particularly in time-series data crucial for financial applications. Unlike traditional sequential processing models, Transformers enable parallel computation and direct modeling of long-range dependencies through their self-attention mechanism, which allows each element in a sequence to attend to all other elements simultaneously [48].

A key strength of Transformer models lies in their ability to process input sequences in parallel, eliminating the sequential bottleneck found in recurrent architectures [48]. This parallel processing capability not only results in significantly faster training times but also enables better handling of long sequences, which is particularly relevant for financial transaction monitoring where patterns may span across extended time periods.

However, the standard Transformer architecture faces computational challenges when dealing with very long sequences, as its self-attention mechanism has quadratic complexity with respect to sequence length [24]. Recent architectural innovations, such as the Reformer, have addressed these efficiency concerns through techniques like locality-sensitive hashing attention and reversible residual connections [24]. These improvements have the potential to make Transformers more practical for real-world applications involving extensive sequential data, such as financial transaction histories.

TranAD [45] is a prominent example of a dedicated transformer network for multivariate anomaly detection. Employing the encoder section of the transformer architecture, TranAD leverages self-attention to capture long-range dependencies within time-series data, enabling the identification of anomalies that manifest over extended periods. Furthermore, it incorporates adversarial training to enhance robustness and improve its ability to generalize to unseen anomaly patterns. Evaluations have shown that TranAD achieves significant performance improvements (up to 17%) compared to traditional methods while drastically reducing training times (up to 99%). This efficiency gain is attributed to the parallel processing capabilities of transformers.

The effectiveness of attention mechanisms for anomaly detection is further highlighted by their successful integration into fraud detection systems [20]. Self-attention mechanisms enable these systems to capture intricate relationships between different features, improving their ability to identify subtle anomalous patterns often missed by conventional methods. This focus on subtle anomalies is directly relevant to AML, where illicit activities are often disguised within seemingly normal transactions. Furthermore, Tatulli et al. [43] presented HAMLET, a hierarchical transformer model for money laundering detection that employs attention mechanisms at both the transaction and sequence levels to effectively capture complex laundering operations carried out through sequences of transactions. Similarly, Busson et al. [7] employed a transformer-based model with a taxonomy-aware attention layer for hierarchical classification of financial transactions. This model, called the Two-headed DragoNet, uses context fusion of transformer-based

embeddings and achieved F1-scores of 93% and 95% on macro-category classification tasks for card and current account datasets, respectively. This demonstrates the effectiveness of transformers in handling hierarchical structures within financial data and improving classification accuracy.

While models like TadGAN [18] and BeatGAN [57] demonstrate the power of combining adversarial training with time-series reconstruction, they rely on RNNs or LSTMs as their backbone. By leveraging transformer encoders, as proposed in this thesis, we aim to capitalize on the advantages of transformers, such as scalability and superior feature extraction, to further improve anomaly detection performance in AML.

3.6 Decision Trees and XGBoost in Financial Fraud Detection

Decision trees and ensemble methods like XGBoost are widely used in financial fraud detection due to their balance of predictive performance and interpretability.

3.6.1 Overview of Applications

Decision trees provide inherent interpretability through their hierarchical structure, making them useful for understanding decision-making processes in fraud detection. XGBoost, which builds upon decision trees through gradient boosting, offers enhanced predictive performance by combining multiple weak learners into a strong ensemble model.

Chen *et al.* [9] demonstrated XGBoost's effectiveness in online transaction fraud detection, showing that it often outperforms other machine learning algorithms when optimized with techniques like the Improved Sailfish Optimizer. Xu *et al.* [53] introduced the Deep Boosting Decision Tree (DBDT) approach, enhancing the interpretability of gradient boosting by integrating neural networks within a decision tree structure. Vassallo *et al.* [47] highlighted XGBoost's adaptability in cryptocurrency transaction fraud detection.

3.6.2 Explainability Considerations

While decision trees offer a degree of inherent explainability, XGBoost models can be less transparent due to their ensemble nature. Techniques like feature importance scores provide some insight, but the complex interactions within boosted trees can obscure the underlying decision-making process.

Priscilla *et al.* [33] showed that optimization techniques can enhance XGBoost's performance without sacrificing the existing level of interpretability provided by the algorithm's feature importance measures. However, achieving full transparency remains a challenge, particularly in high-stakes applications like AML, where clear explanations for decisions are critical.

3.7 Supervised Learning Models for AML

Supervised machine learning models, such as decision trees and support vector machines (SVM), are used for fraud detection and anti-money laundering due to their effectiveness in classification tasks [10]. Decision trees are noted for their simplicity and interpretability, which is beneficial for compliance and auditing, as they allow for straightforward explanations of the decision-making process. Support Vector Machines (SVMs) address this limitation by mapping input features to a higher-dimensional space, enabling linear separation between legitimate and illicit transactions [52]. Despite their strengths, both models can struggle with high-dimensional data and require careful feature engineering.

Random forests extend decision trees by building an ensemble of trees trained on different subsets of the data, improving robustness and predictive performance [52]. Boosted tree methods enhance predictive performance by iteratively adding trees to optimize the difference between predicted and actual values [52]. While these methods often exhibit strong predictive accuracy in AML applications, their complexity can hinder interpretability.

Explainable Boosting Machines (EBMs) offer a compelling alternative by combining accuracy with interpretability [32]. EBMs are a type of Generalized Additive Model (GAM) that learn the relationship between input features and the target variable through a sum of shape functions. These functions can be directly visualized, allowing for clear understanding of feature contributions to predictions. Alahmadi *et al.* [1] demonstrated the application of EBMs in analyzing complex datasets while maintaining interpretability.

3.8 Integration of Unsupervised and Supervised Learning

Combining unsupervised and supervised learning is a powerful approach for fraud detection, capitalizing on the strengths of each.

3.8.1 Challenges in Integration

Integrating unsupervised and supervised methods presents challenges due to their differing learning paradigms. Gomes *et al.* [19] highlighted difficulties in adapting unsupervised deep learning, specifically autoencoders, for insurance fraud detection. Wang *et al.* [49] addressed complexities in leveraging labeled and unlabeled data for fraud detection in multiview networks.

3.8.2 Advantages of Integrated Methods

Despite the challenges, integrated approaches demonstrate significant potential. Shao and Gu [37] combined unsupervised outlier detection with actively learned decision trees, improving both accuracy and training efficiency. Carcillo *et al.* [8] enhanced robustness in credit card fraud detection by incorporating unsupervised outlier detection into supervised learning frameworks. Lacruz and Saniie [26] developed a highly accurate model combining a semi-supervised autoencoder with supervised logistic regression.

3.9 The Need for Explainability in Anti-Money Laundering

Explainability is essential in AML systems to foster trust, ensure compliance, and facilitate effective decision-making.

3.9.1 Regulatory Requirements and Practical Needs

Regulatory bodies emphasize the need for transparency in AI systems used in financial services. Sheth *et al.* [38] stress that transparency is crucial for building trust, particularly when AI systems impact human lives. Bodria *et al.* [6] advocate for transparency by exploring methods for generating counterfactual explanations, offering insights into the decision-making processes of otherwise opaque models.

3.9.2 Explainable Boosting Machines for Interpretability

EBMs offer a balance between accuracy and interpretability, making them suitable for AML applications where understanding the rationale behind risk assessments is

essential. Främling [16] discusses the Contextual Importance and Utility (CIU) method for providing human-like explanations grounded in Multi-Attribute Utility Theory. These techniques enable analysts to understand not just what the model predicts, but why it makes those predictions, facilitating more informed and effective fraud investigations, for example.

3.10 Comparative Analysis

A comparative analysis of existing anomaly detection models reveals various approaches with different strengths and weaknesses for AML. Table 3.1 summarizes VAE-based models and their relevance to AML. Table 3.2 presents transformer-based models, focusing on their ability to capture long-range dependencies. Finally, Table 3.3 highlights other deep learning and hybrid models, outlining their advantages and challenges for AML.

Table 3.1 – VAE-based Anomaly Detection Models for AML

Model	**Key Features and Relevance to Thesis**
VAE [23]	Variational Autoencoders are foundational probabilistic models that learn complex data distributions by encoding data points into a lower-dimensional latent space and then decoding back to the original data space. While effective in learning representations, VAEs can be sensitive to outliers, motivating the development of more robust variants for anomaly detection, such as those employed in AML.
SAVAE [28]	Adversarial training enhances robustness to outliers and noise. Computationally intensive, but central to the unsupervised anomaly detection stage of the proposed framework.
MSCVAE [55]	Employs multi-scale convolutional layers to capture temporal dependencies at various scales, addressing the need for analyzing multi-scale patterns in financial time series, a key aspect of AML.
LSTM-VAE-GAN [31]	Combines LSTM networks and GANs for time series generation, with an emphasis on real-time anomaly detection. Relevant for fast transaction processing in AML, although model complexity can be a concern.
SaVAE-SR [28]	Uses spectral residual preprocessing, enabling unsupervised learning and the effective handling of imbalanced data, directly relevant to AML scenarios. Requires careful parameter tuning.
C-VAE [17]	Leverages chaotic maps to improve the detection of subtle anomalies, particularly effective in imbalanced datasets, which are characteristic of AML. Increased complexity might require more computational resources.

Table 3.2 – Transformer-based Anomaly Detection Models for AML

Model	**Key Features and Relevance to Thesis**
Transformer [48]	The foundational Transformer architecture employs self-attention and parallel processing, enabling efficient capture of long-range dependencies. This capability serves as the basis for specialized transformer models in anomaly detection, although the inherent data requirements and susceptibility to overfitting pose challenges.
AnoFormer [39]	Integrates a two-step masking strategy within a GAN framework for effective anomaly detection and robust representation learning. The complexity of the GAN framework can pose training challenges. Demonstrates the potential of combining transformers and GANs.
TranAD [45]	A dedicated transformer model for multivariate anomaly detection, incorporating adversarial training for performance gains and reduced training times. May require careful hyperparameter tuning. Highlights transformer efficiency for multivariate time series anomaly detection.
Finsformer [3]	Employs a novel cluster-attention mechanism, achieving high precision and recall in financial transaction data. Requires further validation on diverse datasets. Demonstrates the promise of specialized attention mechanisms for financial anomaly detection.
HAMLET [43]	A hierarchical transformer specifically designed for money laundering detection, capturing complex operations through hierarchical attention. Model complexity and the need for hierarchical data are considerations.
TadGAN [18]	Uses GANs for time-series anomaly detection, but with an RNN/LSTM backbone. Powerful for time series reconstruction and benefits from adversarial training, but the RNN/LSTM backbone limits long-range dependency capture. Motivates exploration of transformer-based GANs.
BeatGAN [57]	A bidirectional GAN for time series anomaly detection, also with an RNN backbone. Improves time series reconstruction and captures bidirectional dependencies, but scalability and long-range dependency capture are limited. Further reinforces the potential of transformers for enhanced time series analysis.

Table 3.3 – Other Deep Learning and Hybrid Approaches for AML

Model	**Key Features and Relevance to Thesis**
LSTM Autoencoder [30, 42]	Learns compressed representations of time series data. Captures temporal dependencies and is effective for anomaly detection, but suffers from the vanishing gradient problem and struggles with very long sequences. Serves as a baseline and highlights limitations addressed by transformers.
CNN-BiLSTM [34]	Combines Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory networks (BiLSTMs). Leverages both spatial and temporal features, achieving high accuracy in some fraud detection tasks. Increased model complexity requires careful design. Demonstrates hybrid model effectiveness in related financial applications.
1D Conv-LSTM [54]	Integrates 1D convolutional layers with LSTM. Captures local and temporal patterns, making it suitable for stock market analysis. May not be as effective for very long sequences or complex patterns. Illustrates deep learning's applicability to stock market anomaly detection, relevant to AML.
Deep Reinforcement Learning (DRL) [4]	Employs agent-based learning to learn optimal strategies in dynamic environments. Can outperform supervised methods on large unlabeled datasets, offering potential for AML scenarios with limited labeled data. However, it is difficult to implement, tune, and requires careful reward design.
Graph Neural Networks (GNNs), e.g., NENN [41, 5]	Operate on graph data, capturing relationships between entities. Well-suited for modeling financial transactions as a network, providing a framework for incorporating relational information crucial for understanding AML. Requires graph data representation and is computationally expensive for large graphs.
LSTM-HONN [30]	Combines LSTM with Higher-Order Neural Networks (HONNs) to address stock market unpredictability and improve accuracy. Demonstrates the potential of hybrid models for noisy financial data, but increases complexity and requires HONN expertise.

4. OBJECTIVE AND RESEARCH QUESTIONS

The primary objective of this research is to develop and evaluate a robust and interpretable anomaly detection framework for financial transactions, specifically targeting potential money laundering activities. This framework leverages a novel dual-stage approach, combining a Self-Adversarial Variational Autoencoder (SAVAE) with transformers for unsupervised anomaly detection and an Explainable Boosting Machine (EBM) for supervised classification and interpretability. The framework's performance and generalizability are evaluated using both a proprietary dataset and a publicly available credit card fraud dataset.

This research investigates the following key questions:

1. How effectively does the proposed dual-stage SAVAE-EBM framework detect and classify financial anomalies indicative of potential money laundering in a financial transaction dataset, considering performance metrics and the inherent class imbalance?
2. What insights into the underlying factors contributing to potential money laundering activities can be derived from the EBM's explanations of its classification decisions?
3. How does incorporating transformers within the SAVAE architecture impact the model's ability to capture temporal dependencies and complex patterns relevant to anomaly detection in financial time series?
4. How do the framework's performance and interpretability differ when applied to the publicly available credit card fraud dataset compared to the proprietary dataset?

5. METHODOLOGY

5.1 Introduction

In this chapter, we present the methodology employed to develop an effective and interpretable anomaly detection framework for financial transactions. Our approach integrates a Self-Adversarial Variational Autoencoder (SAVAE) enhanced with transformer blocks and channel-wise attention mechanisms, coupled with an Explainable Boosting Machine (EBM) for classification. This combination leverages deep learning capabilities to capture complex patterns and provides interpretability essential for practical applications in anti-money laundering (AML).

5.2 General Methodology

5.2.1 Model Architecture

Self-Adversarial Variational Autoencoder (SAVAE)

The SAVAE model consists of four main components: an encoder E_ϕ, a decoder D_θ, a discriminator C_ψ, and channel-wise feature attention modules. The architecture is designed to learn robust latent representations of the input data $\mathbf{x} \in \mathbb{R}^d$ and to detect anomalies based on reconstruction errors.

Encoder (E_ϕ)

The encoder transforms the input data into a latent space representation. Its architecture includes:

- **Dense Layers**: Two fully connected layers with scaled exponential linear unit (SELU) activation functions, followed by batch normalization and dropout for regularization.
- **Channel-Wise Feature Attention**: Enhances important features by assigning attention weights through a combination of global average pooling and a multi-layer perceptron (MLP).
- **Transformer Blocks**: Incorporate multi-head self-attention mechanisms to capture dependencies between features. Each block includes layer normalization, self-attention, and a position-wise feed-forward network.

- **Latent Variable Sampling**: Generates the latent variables $\boldsymbol{z}$ using the reparameterization trick:

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{5.1}$$

  where $\boldsymbol{\mu}$ is the mean vector output by the encoder, $\log \boldsymbol{\sigma}^2$ is the log-variance vector output by the encoder, $\boldsymbol{\sigma} = \exp\left(\frac{1}{2} \log \boldsymbol{\sigma}^2\right)$ is the standard deviation, $\odot$ represents element-wise multiplication, and $\boldsymbol{\epsilon}$ is a noise vector sampled from a standard normal distribution.

Decoder (D_θ)

The decoder reconstructs the input data from the latent variables. It mirrors the encoder's architecture with:

- **Dense Layers**: Fully connected layers with SELU activation, batch normalization, and dropout.
- **Channel-Wise Feature Attention**: Applies attention mechanisms to enhance reconstruction quality.
- **Transformer Blocks**: Similar to those in the encoder, to capture feature dependencies during reconstruction.
- **Output Layer**: Produces the reconstructed input using a sigmoid activation function.

Discriminator (C_ψ)

The discriminator aims to distinguish between real data and reconstructed data, promoting the generation of realistic reconstructions. It includes:

- **Flatten Layer**: Converts input data into a one-dimensional array.
- **Dense Layers**: Two fully connected layers with leaky ReLU activation, batch normalization, and dropout.
- **Output Layer**: Produces a probability score using a sigmoid activation function.

Channel-Wise Feature Attention

The attention mechanism assigns weights to different features, enhancing the model's ability to focus on important attributes. It is defined as:

$$\text{Attention}(\mathbf{x}) = \sigma\left(\text{MLP}\left(\text{GAP}(\mathbf{x})\right)\right) \odot \mathbf{x}, \tag{5.2}$$

where σ is the sigmoid activation, MLP is a multi-layer perceptron, GAP denotes global average pooling, and $\odot$ represents element-wise multiplication.

Explainable Boosting Machine (EBM)

The EBM is an interpretable model that combines boosting with additive models. It captures interactions between features while maintaining transparency. The EBM uses gradient boosting on shallow trees and can model feature interactions up to a specified depth.

5.2.2 Loss Functions and Training

The SAVAE model is trained using a composite loss function that balances several objectives:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \alpha\mathcal{L}_{\text{KL}} + \beta\mathcal{L}_{\text{adv}} + \gamma\mathcal{L}_{\text{context}}, \tag{5.3}$$

where:

- $\mathcal{L}_{\text{recon}}$: Reconstruction loss, measured by Mean Squared Error (MSE):

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right]. \tag{5.4}$$

- $\mathcal{L}_{\text{KL}}$: Kullback-Leibler divergence, promoting a smooth latent space:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\sigma_j^2 - \mu_j^2 - \sigma_j^2\right). \tag{5.5}$$

- $\mathcal{L}_{\text{adv}}$: Adversarial loss from the discriminator:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[\log C_\psi(\mathbf{x})\right] + \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\text{model}}}\left[\log\left(1 - C_\psi(\hat{\mathbf{x}})\right)\right]. \tag{5.6}$$

- $\mathcal{L}_{\text{context}}$: Context preservation loss, measured by Mean Absolute Error (MAE):

$$\mathcal{L}_{\text{context}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[\|\mathbf{x} - \hat{\mathbf{x}}\|_1\right]. \tag{5.7}$$

The hyperparameters α, β, and γ are used to balance the contributions of each loss component.

5.2.3 Algorithm Implementation

We present a generalized algorithm for our anomaly detection framework in Algorithm 5.1.

Algorithm 5.1 – Generalized SAVAE-EBM Framework for Anomaly Detection

Input: Dataset X, number of epochs E, anomaly threshold percentile p

Output: Risk classifications for transactions

Split X into training set X_{train} and testing set X_{test}

Data Preprocessing: Apply data cleaning, feature scaling, and normalization to X_{train} and X_{test}

Initialize SAVAE Components: Define encoder E_ϕ, decoder D_θ, and discriminator C_ψ with specified architectures

for *epoch* = 1 **to** E **do**

  for *each batch b in* X_{train} **do**

    $z_{\text{mean}}, z_{\text{log\_var}}, z \leftarrow E_\phi(b)$ $\hat{b} \leftarrow D_\theta(z)$

    Train Discriminator: Compute $\mathcal{L}_{\text{disc}}$ using real data b and generated data $\hat{b}$ Update discriminator weights using $\nabla_\psi \mathcal{L}_{\text{disc}}$

    Train Generator: Compute $\mathcal{L}_{\text{total}}$ using reconstruction, KL divergence, adversarial, and context losses Update encoder and decoder weights using $\nabla_{\phi,\theta} \mathcal{L}_{\text{total}}$

  end

end

Anomaly Detection: Compute reconstruction errors $\text{errors}_{\text{train}}$ on X_{train} Set threshold τ at the p-th percentile of $\text{errors}_{\text{train}}$ $\text{Anomalies}_{\text{SAVAE}} \leftarrow \{\mathbf{x} \in X_{\text{test}} \mid \text{error}(\mathbf{x}) > \tau\}$

Feature Extraction: Extract latent features z_{mean} and $z_{\text{log\_var}}$ for X_{train} and X_{test} Prepare combined feature sets $X_{\text{EBM\_train}}$ and $X_{\text{EBM\_test}}$ including original and latent features

EBM Training: Compute class weights based on $\text{Anomalies}_{\text{SAVAE}}$ in X_{train} Train EBM on $X_{\text{EBM\_train}}$ with class weights

Risk Assignment: $\text{Anomalies}_{\text{EBM}} \leftarrow \text{EBM.predict}(X_{\text{EBM\_test}})$ **for** *each transaction* $\mathbf{x}$ *in* X_{test} **do**

  $A_{\text{SAVAE}} \leftarrow \mathbb{I}(\text{error}(\mathbf{x}) > \tau)$ $A_{\text{EBM}} \leftarrow \text{Anomalies}_{\text{EBM}}[\mathbf{x}]$ Assign risk level R based on A_{SAVAE} and A_{EBM} (see Section 5.2.4)

end

return *Risk classifications for* X_{test}

5.2.4 Risk Scale Definition

We define a three-tier risk classification system based on the outputs of the SAVAE and EBM models:

- **High-Risk Anomaly**: Transactions identified as anomalous by both SAVAE and EBM.
- **Medium-Risk Anomaly**: Transactions identified as anomalous by either SAVAE or EBM.
- **Normal**: Transactions not identified as anomalous by either model.

Formally, the risk level R is defined as:

$$R = \begin{cases} \text{High-Risk Anomaly}, & \text{if } A_{\text{SAVAE}} = 1 \text{ and } A_{\text{EBM}} = 1, \\ \text{Medium-Risk Anomaly}, & \text{if } A_{\text{SAVAE}} \oplus A_{\text{EBM}} = 1, \\ \text{Normal}, & \text{otherwise}, \end{cases} \tag{5.8}$$

where A_{SAVAE} and A_{EBM} are binary indicators of anomalies from the SAVAE and EBM models, respectively, and $\oplus$ denotes the exclusive OR (XOR) operation.

5.3 Application to the Custom Financial Transactions Dataset

This section details the methodology employed in this study, encompassing data collection, feature engineering, and data preprocessing. Specific details and formulas regarding feature engineering and data preprocessing are provided in the Appendix A for reference and are omitted here for brevity and to protect sensitive information regarding specific data characteristics.

5.3.1 Data Collection

The dataset utilized in this study comprises financial transaction records spanning multiple years. To ensure data confidentiality, specific information regarding the data's origin and the timeframe of its collection is not disclosed. The dataset features various transaction attributes, which are described in general terms below.

Dataset Description

The dataset includes financial transaction records, along with contextual data and engineered features. These features can be categorized as follows:

- **Transaction Details:** This category includes fundamental information about each transaction, such as the date and time of the transaction, the transaction amount, the type of asset involved, and the counterparties involved.

- **Client Information:** Client-specific information, including demographic and risk-relevant attributes, further enriches the transactional data. Specific details of these attributes are withheld due to privacy considerations.
- **Engineered Features:** A core component of this study involves constructing features specifically designed to capture potentially suspicious trading behavior. These features are categorized and summarized in the following section. Details of their calculation, including specific formulas and parameter selections, are outlined in Appendix A.

5.3.2 Feature Engineering

We performed extensive feature engineering to capture characteristics of financial transactions and client behavior relevant to detecting money laundering activities. The engineered features are grouped into the following categories:

- **Transaction Volume and Value Metrics:** These features quantify the scale and variability of a client's transactions, including the total value, average value, and the degree of fluctuation in transaction amounts.
- **Day Trading Indicators:** These indicators are designed to identify patterns associated with frequent day trading activities, including trading frequency, the profitability of such trades, and a measure of the consistency of success in day trading.
- **Counterparty Analysis Metrics:** To detect unusual relationships or collusive behavior, metrics related to counterparty interactions were developed. These features capture the concentration of trades with specific counterparties and overall counterparty diversity.
- **Transaction Pattern Indicators:** This category encompasses a range of features aimed at identifying atypical transaction patterns, such as rapid successions of trades, short holding periods, significant price changes, and high-value transactions relative to a client's usual activity.
- **Client Risk Factors:** Client-specific risk factors, such as age, geographic location, and financial capacity, were also incorporated as features to contextualize transactions within client profiles.
- **Unusual Activity Indicators:** These binary flags highlight unusual or suspicious activities, such as after-hours trading, asset concentration, circular trading, and unusual trading volume relative to established norms or the client's typical behavior.

- **Additional Alerts and Risk Indicators:** Further indicators were constructed to detect patterns associated with specific money laundering techniques, like structuring and round number transactions, and unusual patterns in counterparty diversity and transaction times.
- **Swing Trading Indicators:** This set of features captures the characteristics of swing trades, where assets are held for multiple days, considering aspects like the number and frequency of such trades and their typical holding periods.

5.3.3 Data Preprocessing

Before model training, the dataset was preprocessed to improve data quality and model performance. The preprocessing steps include:

- **Data Cleaning and Imputation:** Missing values were handled using appropriate imputation techniques, including median imputation and zero-filling, depending on the nature and context of each feature. Duplicate records and irrelevant features were removed to enhance data quality.
- **Feature Scaling and Normalization:** To ensure that features with different scales contribute equitably to the model training process, features were scaled using Min-Max normalization to a common range. Values deemed infinite as a result of calculations (e.g., due to division by zero) were replaced with the median value of the respective feature.
- **Correlation Analysis and Feature Selection:** To mitigate multicollinearity, we performed a correlation analysis to identify and manage highly correlated features. We removed one feature for each pair of features exhibiting high correlation, preferentially retaining aggregate metrics. This feature selection approach minimized redundancy within the feature set while preserving crucial informational content.

5.3.4 Model Training and Evaluation

SAVAE Model Training

The SAVAE model was trained using the preprocessed training data with the following hyperparameters:

Table 5.1 – SAVAE Hyperparameter Configuration for Custom Dataset

Parameter	Value
Latent Dimension	24
Intermediate Dimension	256
Number of Transformer Blocks	3
Number of Attention Heads	8
Feed-Forward Dimension	128
Learning Rate	3×10^{-4}
Batch Size	2048

We selected the hyperparameters in Table 5.1 through ad hoc experiments and empirical testing, evaluating various SAVAE architecture configurations and parameter settings. We identified the values that optimized performance by iteratively testing different model variations and adjusting key parameters. Both experimental results and domain knowledge guided this approach, though it was constrained by limited computational resources and the inherently subjective nature of empirical evaluations. The model was trained for 20 epochs, employing early stopping and learning rate reduction based on validation performance to prevent overfitting and enhance training efficiency.

Anomaly Detection

We identify anomalies by setting the threshold at the 99.5th percentile of the training reconstruction error distribution. Through ad hoc experiments, we selected this threshold to address the unlabeled nature of our private dataset, opting for a more inclusive cutoff than the 99.828th percentile corresponding to the 0.172% class imbalance observed in [46]. This broader threshold deliberately captures more potential anomalies, knowing that our subsequent risk classification process, which designates a transaction as high risk only when identified by both the SAVAE and EBM models, will provide a more precise and selective final assessment.

EBM Training

We prepared the combined feature set by merging the original features with SAVAE-derived latent features (μ and $\log \sigma^2$). To address class imbalance, we computed class weights inversely proportional to class frequencies. We selected the EBM hyperparameters through ad hoc experiments and prior tests using random search, refining them iteratively to optimize performance. We trained the EBM using the following parameters:

- **Interaction Depth**: 4
- **Learning Rate**: 0.1565

- **Maximum Bins**: 151
- **Maximum Rounds**: 435
- **Minimum Samples per Leaf**: 1

Results and Evaluation

We evaluated the model's performance using metrics such as accuracy, precision, recall, F1 score, ROC AUC, and PR AUC. The models demonstrated effectiveness in identifying anomalous transactions.

Interpretability and Insights

Global and local interpretability reports were generated using the EBM's explainability features. The most influential features were identified, providing insights into the factors contributing to anomaly detection.

5.4 Application to the Credit Card Fraud Detection Dataset

5.4.1 Dataset Description

The Credit Card Fraud Detection dataset from Kaggle [46] contains transactions made by European credit cardholders over two days in September 2013. It includes 284,807 transactions, with 492 fraud cases (approximately 0.172%).

Data Structure

The dataset comprises:

- **Features**: 28 principal components (`V1` to `V28`), `Time`, and `Amount`.
- **Target Variable**: `Class`, where 1 indicates a fraudulent transaction.

5.4.2 Data Preprocessing

Normalization

The `Time` and `Amount` features were scaled using min-max normalization, fitting the scaler only on the training data to prevent data leakage.

Handling Imbalanced Data

Due to the highly imbalanced nature of the dataset, we employed techniques such as class weighting and threshold adjustment during model training and evaluation.

5.4.3 Model Training and Evaluation

SAVAE Model Training

The SAVAE model was trained on the preprocessed training data. The anomaly threshold was set at the 99.9th percentile of the training reconstruction error distribution due to the dataset's imbalance.

EBM Training

The EBM was trained on the combined feature set, including the original features and SAVAE-derived latent features. Class weights were calculated to address class imbalance.

Results and Evaluation

Model performance was evaluated using the same metrics as for the custom dataset. The models effectively identified fraudulent transactions, demonstrating the methodology's applicability to different datasets.

6. RESULTS

6.1 SAVAE Model Training and Analysis

6.1.1 Training Convergence

The Self-Adversarial Variational Autoencoder (SAVAE) demonstrated robust training performance on a substantial dataset comprising 867,925 training samples and 216,982 testing samples, with 35 preprocessed features. The model's training history, illustrated in Figure 6.1, exhibits distinct convergence patterns across multiple loss components. The discriminator loss stabilized rapidly at approximately 1.4, while the generator adversarial loss showed gradual convergence to 0.7, indicating effective adversarial training. The reconstruction and context losses, along with their validation counterparts, demonstrated consistent minimization patterns, suggesting robust generalization capabilities.

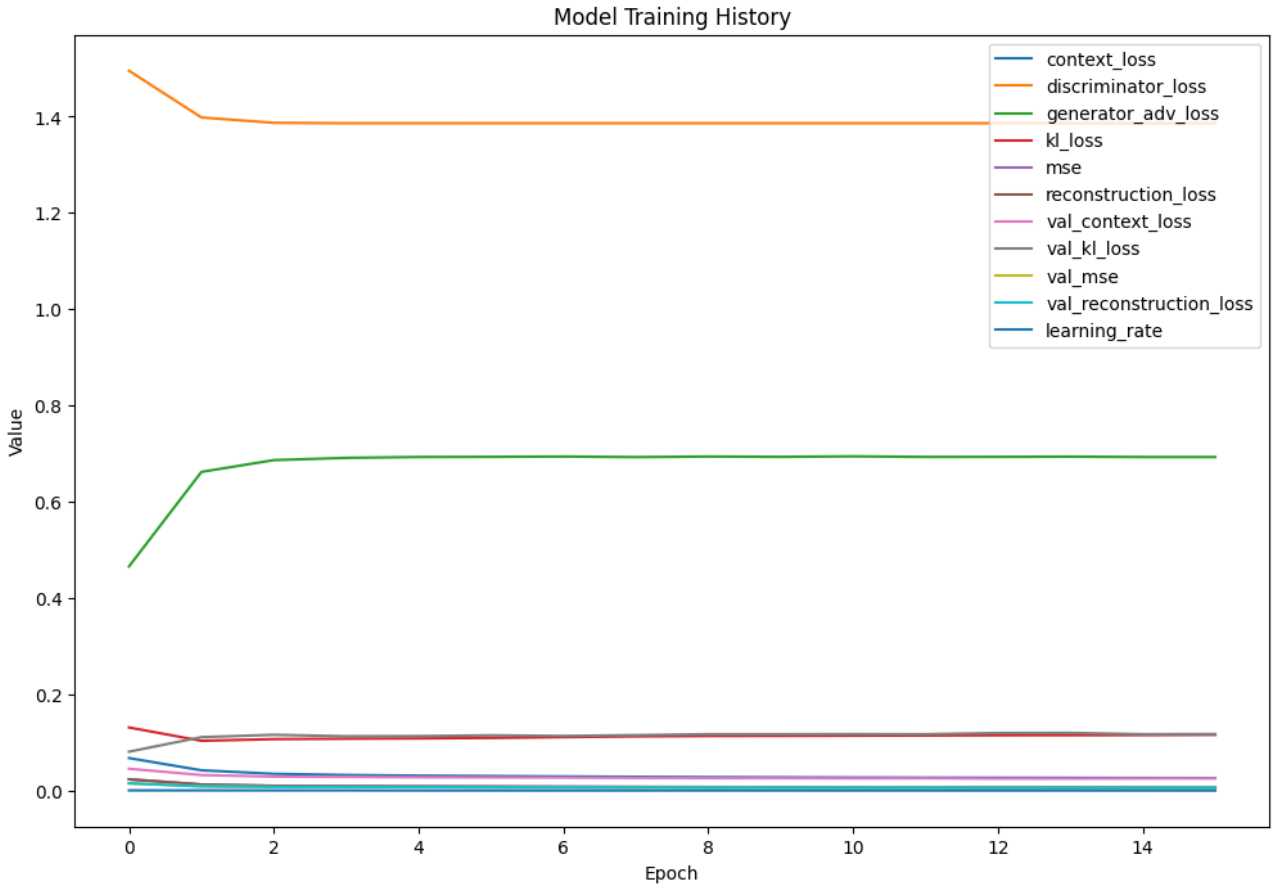

Figure 6.1 – Training history showing the convergence of multiple loss components including reconstruction loss, KL divergence, adversarial loss, and context loss. The close alignment between training and validation metrics indicates effective generalization.

6.1.2 Latent Space Analysis

The model's latent space representation, visualized through t-SNE dimensionality reduction in Figure 6.2, reveals sophisticated pattern organization across 50,000 samples. The visualization demonstrates clear structural differentiation between normal transactions (shown in purple) and anomalous patterns (indicated by yellow highlights). Normal transactions form coherent manifolds with distinct clustering patterns, while anomalous transactions appear predominantly at cluster boundaries or in isolated regions. This spatial organization suggests effective feature learning and pattern discrimination capabilities.

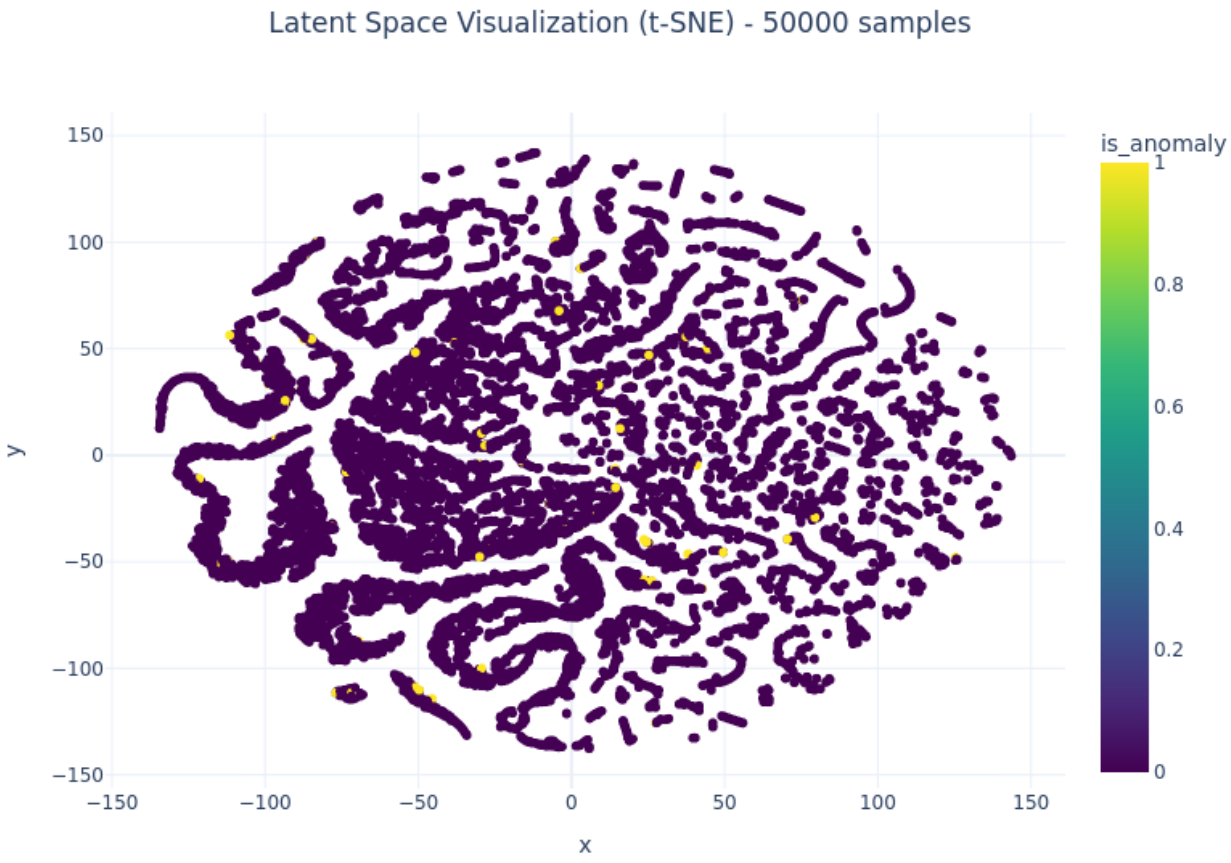

Figure 6.2 – t-SNE visualization of the latent space representation for 50,000 samples. The distinct organization of normal transactions (purple) and anomalous patterns (yellow) demonstrates effective pattern differentiation.

6.1.3 Reconstruction Error Analysis

The distribution of reconstruction errors provides crucial insights into the model's discriminative capabilities. Figure 6.3 presents a comparative analysis of reconstruction errors between training and testing sets. The distributions exhibit consistent patterns, with both sets showing a pronounced right-skewed shape and clear separation between normal and anomalous transactions. The 99.5th percentile threshold, established at 0.043177, effectively delineates the boundary between normal and anomalous patterns.

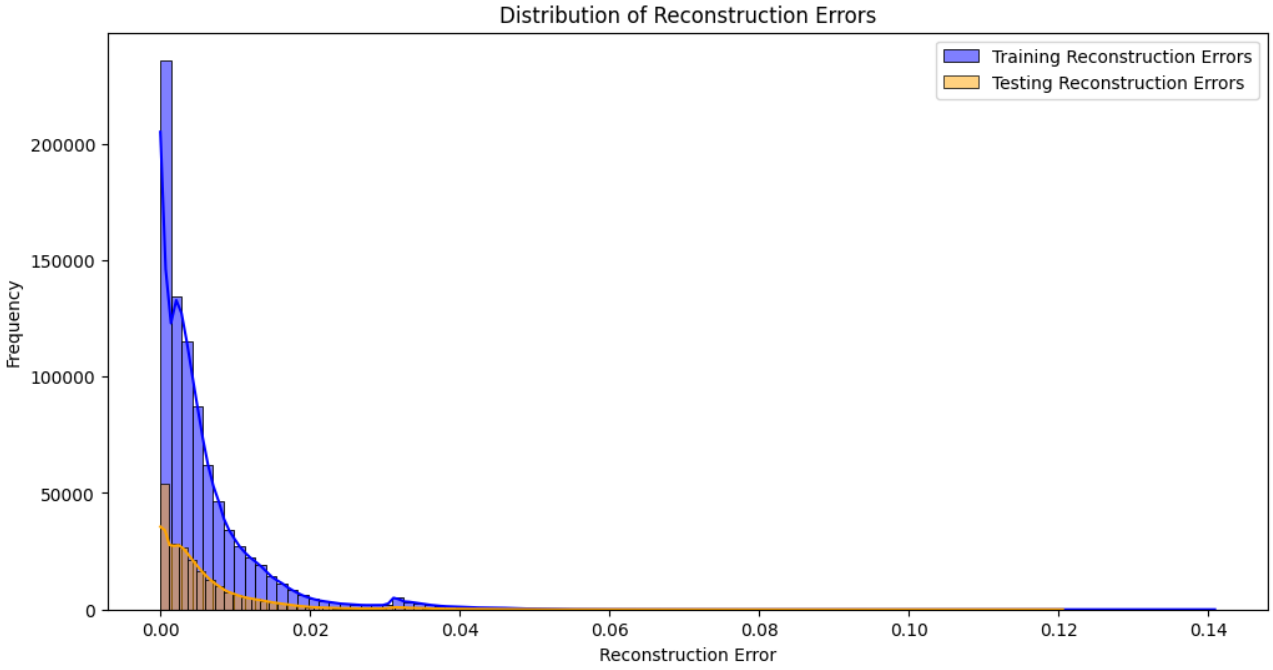

Figure 6.3 – Distribution of reconstruction errors for training and testing sets. The consistent patterns and clear separation demonstrate the model's stable learning and generalization capabilities.

6.1.4 Feature-wise Reconstruction Analysis

Analysis of reconstruction quality across features reveals hierarchical patterns of importance, as shown in Figure 6.4. Features with indices 17-20 demonstrate higher reconstruction errors, suggesting their particular significance in anomaly detection. This gradual progression of error magnitudes shows the model's ability to identify and prioritize distinctive transaction characteristics effectively.

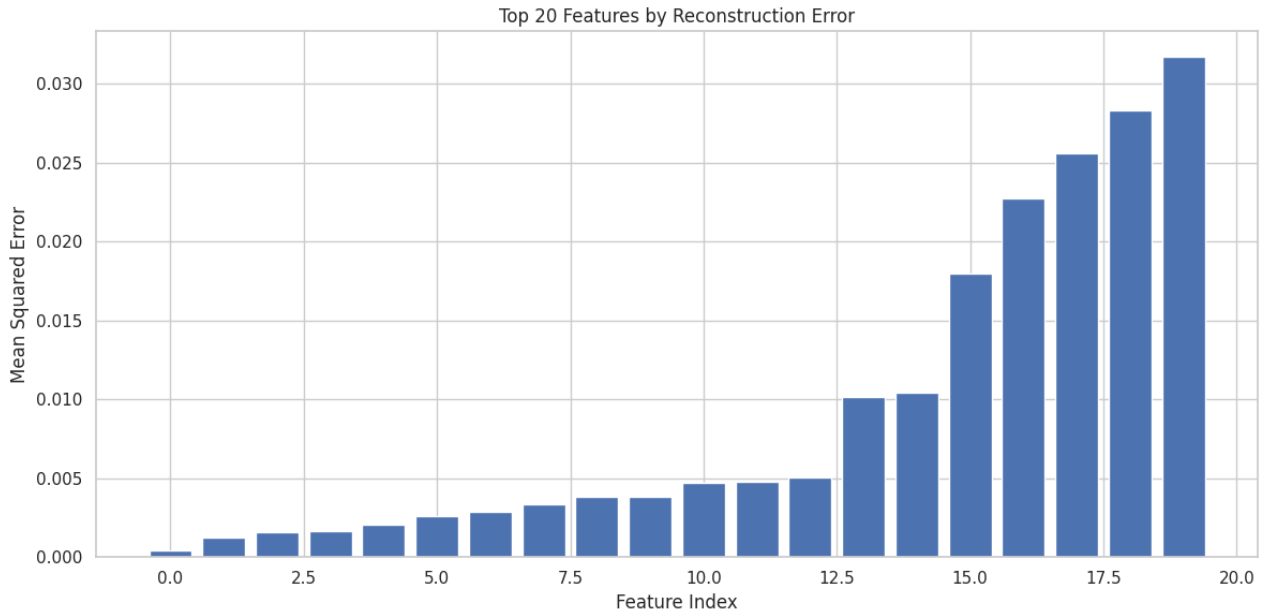

Figure 6.4 – Feature-wise reconstruction error analysis displaying the top 20 features with highest reconstruction error, indicating hierarchical feature importance in anomaly detection.

6.2 Model Performance Evaluation

6.2.1 Classification Performance

The combined SAVAE-EBM framework demonstrated promising anomaly detection capabilities, particularly given the significant class imbalance. While achieving 99.55% accuracy on the test set, a metric known to be less informative with imbalanced data, the model yielded a Matthews Correlation Coefficient (MCC) of 0.5263, a more robust measure in such scenarios. The ROC AUC of 0.9508 and PR AUC of 0.5417 further support the model's discriminative potential.

6.2.2 ROC and Precision-Recall Analysis

The model's discriminative capabilities are comprehensively illustrated through the ROC and Precision-Recall curves shown in Figure 6.5. The ROC curve demonstrates excelent performance with an AUC of 0.9508, with particularly strong performance in the critical low false-positive rate region. The Precision-Recall curve, with an AUC of 0.5417, reflects the model's robust performance despite the extreme class imbalance inherent in financial anomaly detection.

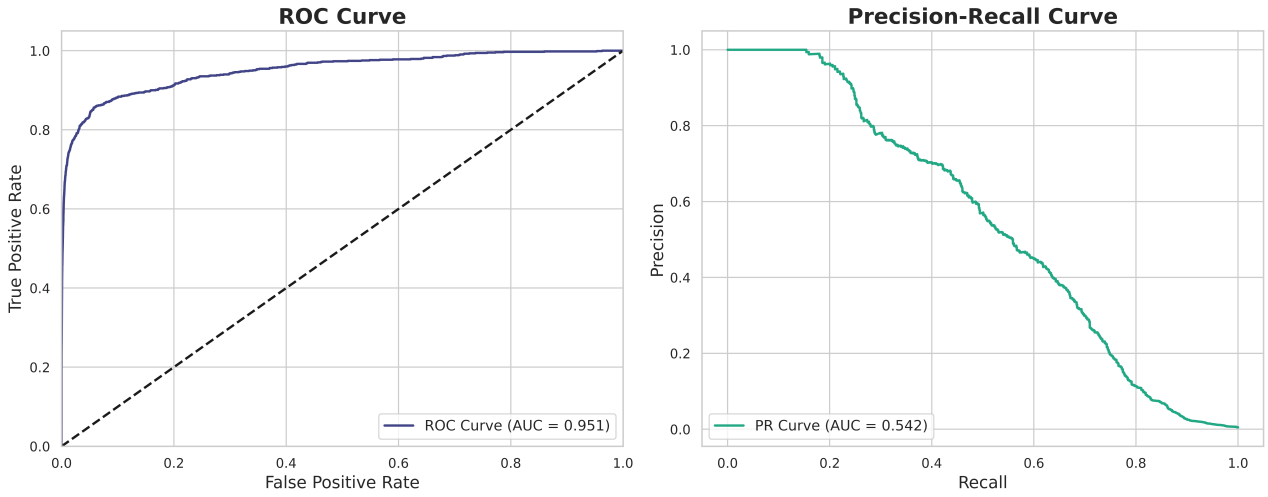

Figure 6.5 – Performance curves demonstrating the model's discriminative capabilities. Left: ROC curve showing strong discrimination (AUC = 0.9508). Right: Precision-Recall curve reflecting robust performance under class imbalance (AUC = 0.5417).

6.2.3 Threshold Optimization and Performance Trade-offs

The selection of an appropriate classification threshold represents a critical operational decision in anomaly detection systems. Figure 6.6 illustrates the intricate relationships between precision, recall, and F1-score across different threshold values.

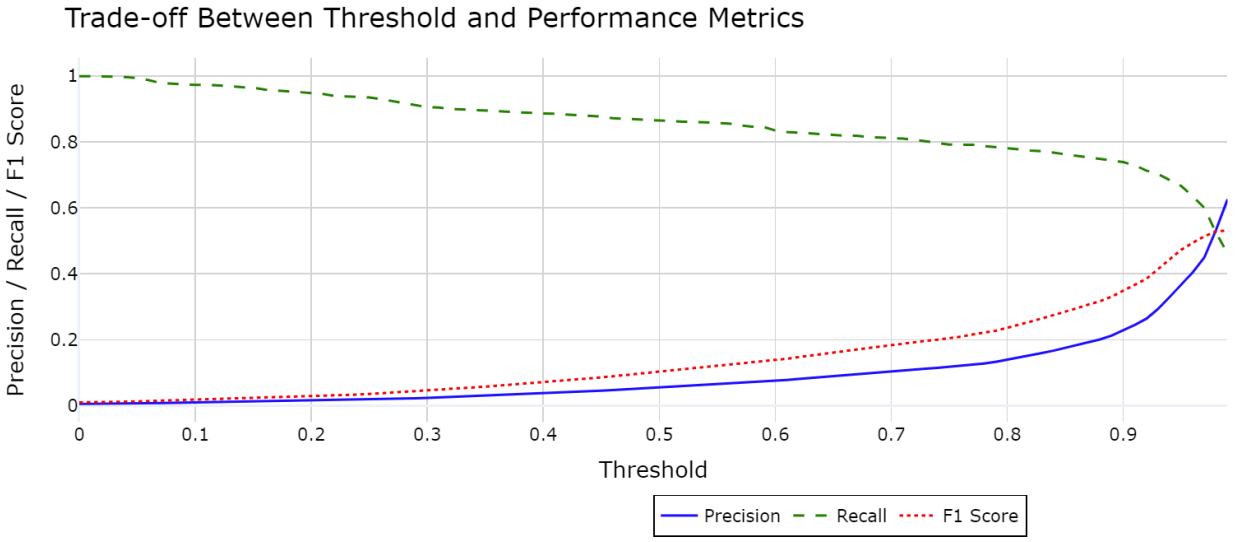

Figure 6.6 – Trade-off analysis between threshold values and performance metrics. The graph demonstrates the inverse relationship between precision and recall, with the F1-score providing a balanced measure. The chosen threshold of 0.98 optimizes the balance between false positives and detection capability.

The analysis reveals several critical trade-off patterns:

- At lower thresholds (0.0-0.3), the model maintains high recall (>0.9) but suffers from poor precision (<0.1), indicating excessive false positives
- The middle range (0.3-0.7) shows gradual improvement in precision with a corresponding decrease in recall
- At higher thresholds (0.7-1.0), precision increases more rapidly, while recall declines at a moderate rate

Based on this analysis, we selected an optimal threshold of 0.98, which achieves several operational objectives:

- Maximizes precision (0.5324) while maintaining acceptable recall (0.5248)
- Results in an F1-score of 0.5285, representing a balanced trade-off between precision and recall
- Produces a manageable number of alerts for investigation, with 551 high-risk anomalies (0.25% of transactions)

This threshold choice is particularly justified for financial anomaly detection, where false positives can be operationally costly. The selected threshold ensures that flagged transactions have a higher probability of being genuine anomalies, while still maintaining sufficient sensitivity to detect suspicious patterns. The resulting performance metrics at this threshold, including an accuracy of 99.55% and MCC of 0.5263, validate the effectiveness of this selection for practical deployment.

6.2.4 Risk Level Distribution

The model's risk assessment framework produced a nuanced three-tier classification system, with the following distribution in the test set:

Table 6.1 – Risk Level Distribution in Test Set

Risk Level	**Count**	**Percentage**
Normal	215,448	99.29%
Medium Risk Anomaly	983	0.45%
High Risk Anomaly	551	0.25%

This distribution demonstrates the model's ability to maintain high precision while identifying a manageable number of high-risk cases for detailed investigation.

6.3 Model Interpretability Analysis

6.3.1 Global Feature Importance

The analysis of global feature importance, visualized in Figure 6.7, reveals a hierarchical structure of predictive features. Geographic risk factors emerged as significant predictors in the model's decision-making process, along with trading pattern indicators. The importance scores demonstrate effective integration of both traditional risk indicators and learned latent representations.

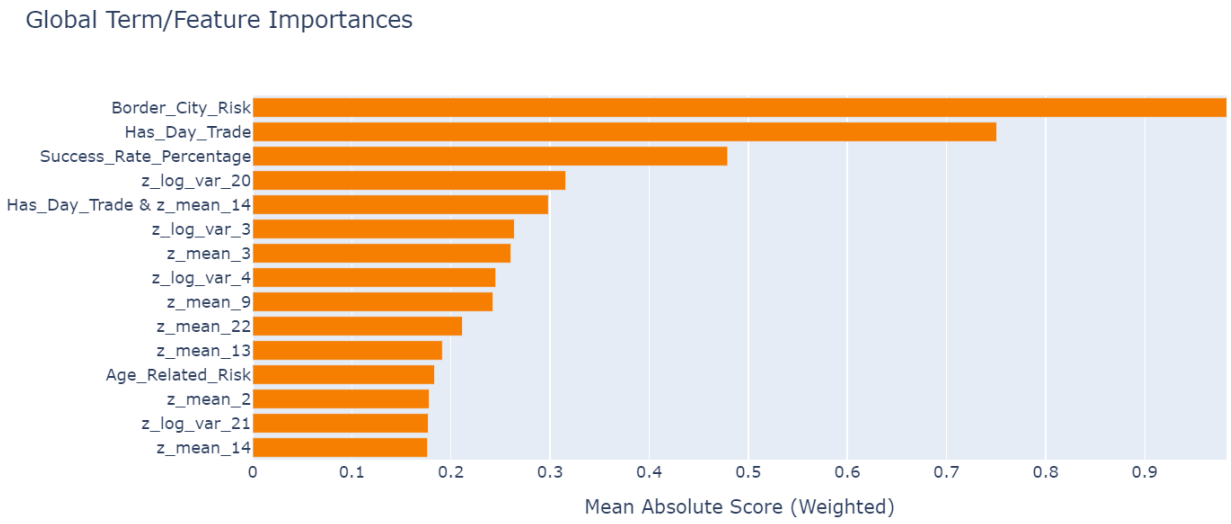

Figure 6.7 – Global feature importance analysis showing the relative contribution of different features to anomaly detection. The hierarchical importance structure reveals the effective combination of domain-specific indicators and learned representations.

6.3.2 Local Explanation Analysis

Individual prediction analysis provides crucial insights into the model's decision-making process, as demonstrated in Figure 6.8. For a high-confidence prediction (Pr(y=1): 0.998), the local explanation reveals complex feature interactions, with Border City Risk and Cross Market Trades providing strong positive contributions to the anomaly classification.

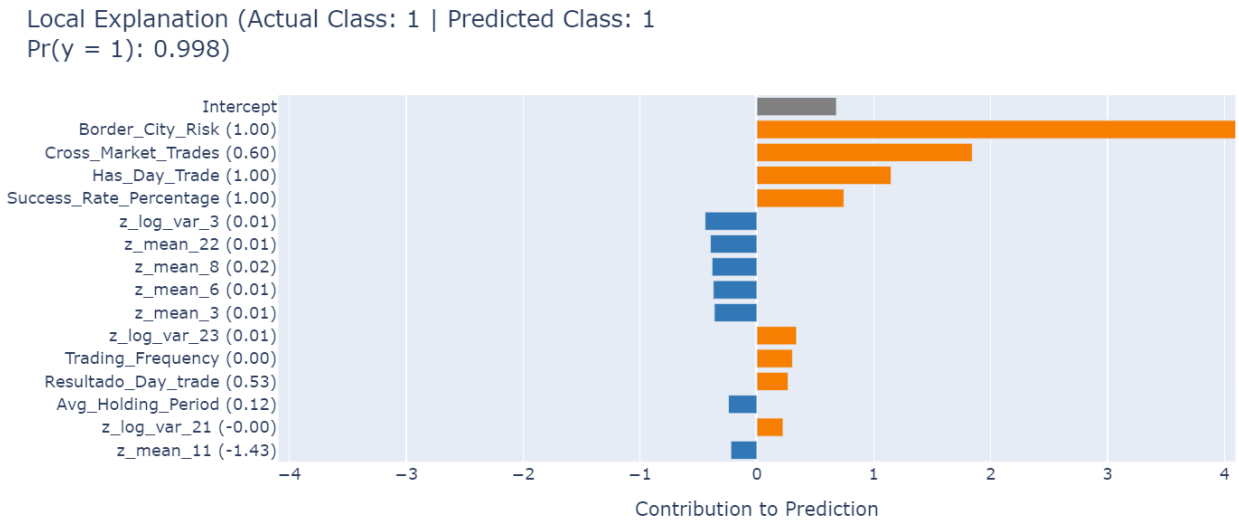

Figure 6.8 – Local feature contributions for a high-confidence anomaly prediction, demonstrating the interplay between features in determining the final classification decision.

6.4 Detailed Performance Analysis

6.4.1 Classification Metrics

The model demonstrated strong performance across multiple evaluation metrics:

Table 6.2 – Comprehensive Performance Metrics

Metric	**Value**
Accuracy	99.55%
Matthews Correlation Coefficient	0.5263
F1-Score	0.5285
ROC AUC	0.9508
PR AUC	0.5417
Precision	0.5324
Recall	0.5248

6.4.2 Operational Implications

The model's performance characteristics suggest strong potential for practical deployment in financial monitoring systems. The high ROC AUC (0.9508) indicates strong discriminative ability, while the precision-recall balance (PR AUC: 0.5417) demonstrates

robust performance under real-world conditions. The three-tier risk classification system, with only 0.25% of transactions classified as high-risk anomalies, provides an operationally manageable framework for investigation prioritization.

The fusion of SAVAE-generated latent features with traditional risk indicators creates a comprehensive anomaly detection framework that balances automated pattern discovery with domain expertise. The clear separation in reconstruction error distributions and robust classification metrics validate the effectiveness of this hybrid approach for financial anomaly detection.

6.5 Model Robustness and Preprocessing Analysis

The preprocessing phase ensured robust feature engineering, handling 35 distinct features including both binary and continuous variables. The careful treatment of correlated features and standardization procedures contributed to the model's stability. The consistent performance between training and testing sets, particularly in reconstruction error distributions, validates the robustness of the preprocessing pipeline and the model's generalization capabilities.

6.6 Evaluation on Public Credit Card Fraud Detection Dataset

To assess the generalization capabilities of our SAVAE-EBM framework, we evaluated its performance on the Credit Card Fraud Detection dataset (described in Section 5.4), which presents a significant class imbalance challenge (0.172% fraudulent transactions).

6.6.1 Model Performance Analysis

The SAVAE-EBM framework achieved a ROC AUC of 0.964 on the test set using original labels, demonstrating strong discriminative power despite operating in an unsupervised manner during training. This performance is particularly noteworthy as the model relies on pseudo-labels generated by the SAVAE component rather than true fraud labels during the training phase. Figure 6.9 illustrates the model's performance through ROC and Precision-Recall curves, with the PR AUC of 0.532 reflecting the framework's effectiveness in handling extreme class imbalance.

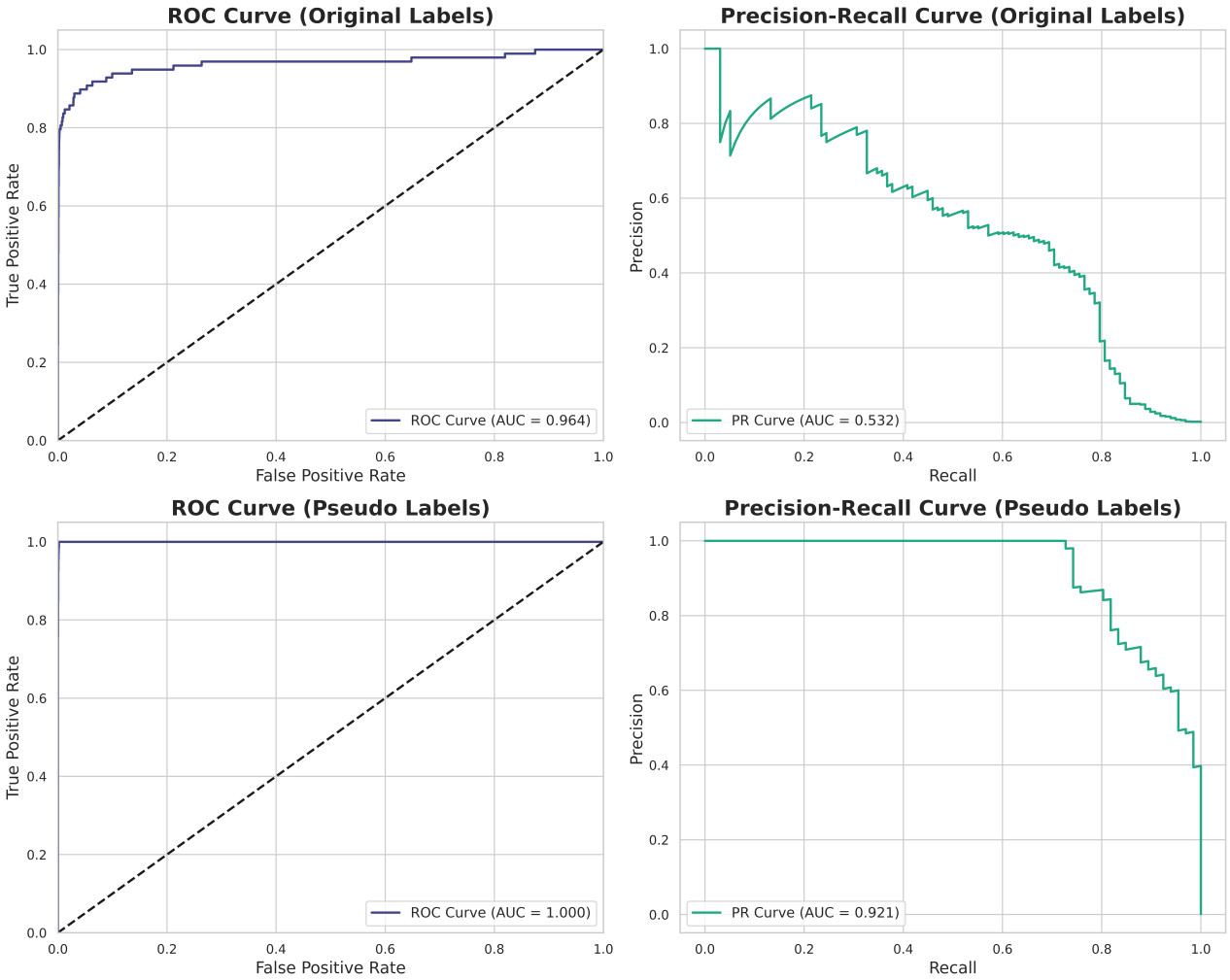

Figure 6.9 – ROC and Precision-Recall curves for the SAVAE-EBM model evaluated on original labels. The high ROC AUC (0.964) demonstrates strong discriminative power, while the PR AUC (0.532) reflects the challenges of extreme class imbalance.

6.6.2 Threshold Analysis and Operational Regimes

Our analysis revealed three distinct operational regimes, each offering specific advantages for different deployment scenarios. In the high-sensitivity regime ($t \leq 0.01$), the model achieves exceptional recall of 94.9% at $t = 0.0001$, maintaining strong recall of 83.7% even at $t = 0.01$. This configuration proves particularly valuable for regulatory compliance scenarios where missing fraudulent transactions carries significant risk.

The balanced performance regime ($0.01 < t \leq 0.5$) represents an optimal configuration for operational deployment under resource constraints. At $t = 0.5$, the model achieves 40.3% precision while maintaining 74.5% recall, with an F1-score of 52.3%. This balance between precision and recall enables efficient allocation of investigative resources.

In the high-precision regime ($t > 0.5$), the model achieves peak precision of 60.9% at $t = 0.99$ while maintaining 42.9% recall. This configuration is particularly suitable for scenarios requiring high confidence in fraud predictions, where false positives carry significant operational costs.

6.6.3 Comparative Performance Analysis

Table 6.3 presents a comprehensive comparison with state-of-the-art methods in credit card fraud detection. Our SAVAE-EBM framework demonstrates competitive performance across various operating points, particularly noteworthy given its unsupervised training approach.

Table 6.3 – Performance Comparison with State-of-the-Art Methods

Method	**Precision**	**Recall**	**F1-Score**	**ROC AUC**	**PR AUC**
SAVAE-EBM (t=0.0001)	0.011	**0.949**	0.023	**0.964**	**0.532**
SAVAE-EBM (t=0.01)	0.109	0.837	0.193	**0.964**	**0.532**
SAVAE-EBM (t=0.1)	0.262	0.796	0.394	**0.964**	**0.532**
SAVAE-EBM (t=0.5)	0.403	0.745	0.523	**0.964**	**0.532**
SAVAE-EBM (t=0.99)	0.609	0.429	0.503	**0.964**	**0.532**
UAAD-FDNet [20]	**0.980**	0.755	**0.853**	0.952	-
Autoencoder+Clustering [56]	0.116	0.816	0.204	0.961	-
Deep Autoencoder [21]	-	0.903	-	0.882	-

6.6.4 Performance Trade-off Analysis

Figure 6.10 illustrates the relationship between different threshold values and performance metrics. This analysis reveals the model's flexibility in adapting to various operational requirements through threshold adjustment.

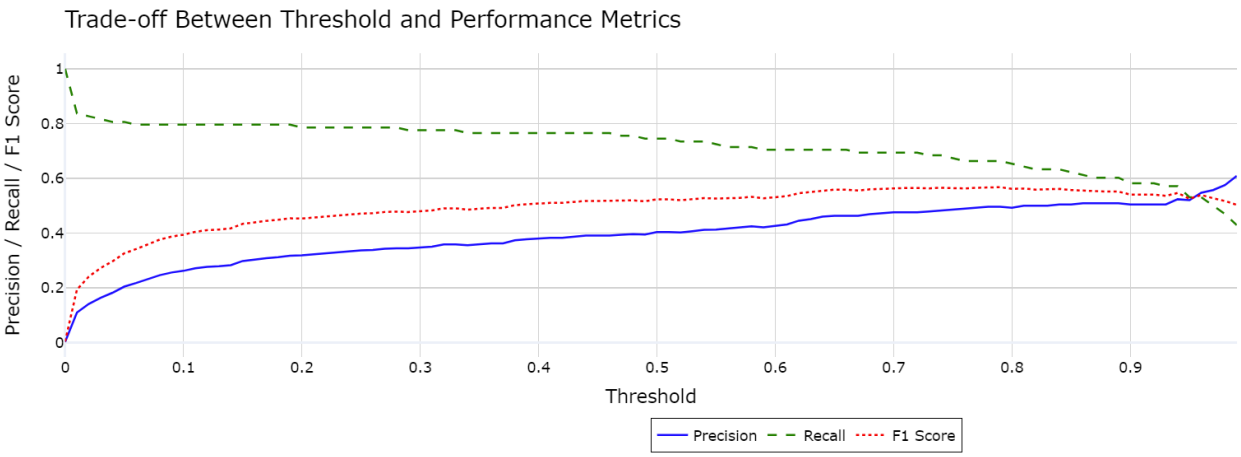

Figure 6.10 – Performance metric trade-offs across different threshold values evaluated on original labels. The plot demonstrates the model's ability to achieve various operating points suitable for different business requirements.

6.6.5 Practical Implications

The framework's performance on the public dataset demonstrates several significant capabilities. Most notably, the model achieves competitive results without requiring labeled training data during the learning phase, as evidenced by the ROC AUC of 0.964 that surpasses several approaches in the literature. The ability to maintain consistent performance across different threshold settings provides operational flexibility, with the high-sensitivity regime achieving 94.9% recall at t=0.0001, the balanced regime reaching an F1-score of 52.3% at t=0.5, and the high-precision regime attaining 60.9% precision at t=0.99.

7. CONCLUSION

Financial anomaly detection presents unique challenges that demand both sophisticated pattern recognition and transparent decision-making processes. This thesis addresses these challenges through a novel dual-stage framework combining a Self-Adversarial Variational Autoencoder (SAVAE) enhanced with transformer blocks and an Explainable Boosting Machine (EBM) while maintaining regulatory compliance through interpretable results.

This MSc advances the capture of temporal patterns in financial data by integrating transformer blocks within the SAVAE architecture. Channel-wise attention mechanisms and self-adversarial training work in concert with these transformer blocks to learn comprehensive representations of normal transaction patterns. Unlike traditional supervised approaches to fraud detection, our framework operates effectively without requiring labeled training data.

7.1 Addressing Research Questions

This work successfully addressed its four central questions:

RQ1: How effectively does the proposed dual-stage SAVAE-EBM framework detect and classify financial anomalies indicative of potential money laundering in a financial transaction dataset, considering performance metrics and the inherent class imbalance?

The framework demonstrated robust performance across multiple evaluation metrics. On the proprietary dataset, the model achieved a ROC AUC of 0.9508 and PR AUC of 0.5417, indicating strong discriminative capability despite significant class imbalance. The three-tier classification system identified 0.25% of transactions as high-risk anomalies, providing a manageable volume for investigation while maintaining high precision (0.5324) and recall (0.5248).

RQ2: What insights into the underlying factors contributing to potential money laundering activities can be derived from the EBM's explanations of its classification decisions?

The EBM component revealed hierarchical relationships among features, with certain location-based risk factors and transaction characteristics emerging as significant predictors. Global feature importance analysis showed effective integration of pre-existing risk indicators and newly learned representations. Local explanations provided insights

into feature interactions, particularly highlighting how combinations of geographic and transaction-related features contributed to high-confidence predictions.

RQ3: How does incorporating transformers within the SAVAE architecture impact the model's ability to capture temporal dependencies and complex patterns relevant to anomaly detection in financial time series?

The integration of transformer blocks and channel-wise attention within the SAVAE architecture contributes to the model's overall ability to learn and represent complex patterns in financial time series. This is clearly observed in the improved separation of normal and anomalous transactions within the latent space of the proprietary dataset, as visualized using t-SNE in Figure 6.2, and the distinct reconstruction error distributions in Figure 6.3. Furthermore, the model's strong performance on the public credit card fraud dataset provides additional, albeit indirect, support for the effectiveness of incorporating transformers.

RQ4: How do the framework's performance and interpretability differ when applied to the publicly available credit card fraud dataset compared to the proprietary dataset?

The framework demonstrated strong generalization capabilities, achieving a ROC AUC of 0.964 on the public credit card fraud dataset, surpassing recent approaches in the literature. Notably, it maintained robust performance across different operational thresholds, achieving 94.9% recall at t=0.0001 in high-sensitivity settings, and 60.9% precision with 42.9% recall at t=0.99 in high-precision settings. This performance was achieved without requiring labeled training data, demonstrating the framework's potential for broad application across different financial contexts.

These findings offer practical value for financial institutions implementing AML systems. Operating without labeled training data, the framework's unsupervised learning component readily adapts to emerging patterns while meeting regulatory requirements through interpretable decisions. The three-tier classification system enables strategic allocation of investigative resources, with clear prioritization of high-risk transactions.

7.2 Limitations

While this research presents advances in financial anomaly detection, several important limitations should be acknowledged. First, while effective, the framework's reliance on reconstruction errors for anomaly detection may not capture all types of financial crime patterns, particularly those that closely mimic legitimate transaction behaviors. The

model's effectiveness is inherently bounded by the quality and representativeness of the normal transaction patterns it learns during training.

Second, while the EBM component provides valuable interpretability, its explanations are primarily feature-centric and may not fully capture complex interactions between multiple features that sophisticated money laundering schemes might exploit. The current implementation's ability to explain decisions is limited to individual feature contributions and may not fully elucidate the temporal aspects of detected anomalies.

Third, the framework's evaluation, though comprehensive, was conducted on datasets with specific characteristics and time periods. The model's generalizability to significantly different financial contexts or transaction types remains to be fully established. Additionally, the extreme class imbalance in the public dataset (0.172% fraudulent transactions) may not perfectly represent the distribution of all types of financial crimes in different contexts.

Fourth, while the unsupervised learning component reduces the need for labeled training data, this same characteristic makes it challenging to definitively validate the model's effectiveness against novel, previously unseen types of financial crime. The framework's ability to adapt to emerging fraud patterns, while promising, requires further longitudinal studies to fully validate.

Fifth, the model exhibits significant performance trade-offs at different threshold settings, as demonstrated in the operational regime analysis. While achieving high recall (94.9%) at lower thresholds (t=0.0001) and high precision (60.9%) at higher thresholds (t=0.99), maintaining balanced performance across both metrics remains challenging. This necessitates careful threshold tuning based on specific operational requirements and risk tolerances, which may need frequent adjustment as financial crime patterns evolve.

7.3 Future Work

Several promising research directions emerge from this work that could further enhance the framework's capabilities and address its current limitations.

First, advancing the model's interpretability mechanisms presents a crucial area for development. While the current EBM implementation provides valuable insights, future work could explore hierarchical interpretability frameworks that explain decisions at multiple levels of abstraction, from individual transaction features to broader pattern recognition. This could include developing visualization techniques specifically designed for temporal patterns in financial data, making complex transformer-based decisions more accessible to domain experts.

The incorporation of semi-supervised learning approaches presents another valuable direction for research. As suspicious transactions are investigated and labeled over

time, these confirmed cases could be integrated into the model's training process. This could involve developing adaptive learning mechanisms that maintain the framework's unsupervised advantages while leveraging newly available labeled data to refine detection capabilities.

Exploring the integration of domain knowledge into the model architecture itself offers another promising avenue. This could involve developing specialized attention mechanisms that incorporate known patterns of suspicious behavior, or designing new loss functions that better reflect the priorities of financial crime detection. Research into methods for incorporating regulatory requirements and domain expertise directly into the model's architecture could enhance both performance and practical utility.

The challenge of extreme class imbalance in financial fraud detection, exemplified by the public dataset where fraudulent transactions comprise only 0.172% of total transactions, remains an area for innovation. Future work could investigate novel approaches to synthetic data generation that preserve privacy while creating realistic examples of suspicious patterns. Additionally, developing methods to better capture the evolution of fraudulent behaviors over time could improve the model's ability to detect emerging patterns of financial crime.

REFERENCES

[1] Alahmadi, R.; Almujibah, H.; Alotaibi, S.; Elshekh, A. E. A.; Alsharif, M.; Bakri, M. “Explainable boosting machine: A contemporary glass-box model to analyze work zone-related road traffic crashes”, *Safety*, vol. 9–4, 2023.

[2] Alarab, I.; Prakoonwit, S.; Nacer, M. I. “Comparative analysis using supervised learning methods for anti-money laundering in bitcoin”. In: Proceedings of the 2020 5th International Conference on Machine Learning Technologies, 2020, pp. 11–17.

[3] An, H.; Ma, R.; Yan, Y.; Chen, T.; Zhao, Y.; Li, P.; Li, J.; Wang, X.; Fan, D.; Lv, C. “Finsformer: A novel approach to detecting financial attacks using transformer and cluster-attention”, *Applied Sciences*, vol. 14–1, 2024.

[4] Arshad, K.; Ali, R. F.; Muneer, A.; Aziz, I. A.; Naseer, S.; Khan, N. S.; Taib, S. M. “Deep reinforcement learning for anomaly detection: A systematic review”, *IEEE Access*, vol. 10, 2022, pp. 124017–124035.

[5] Assumpção, H. S.; Souza, F.; Campos, L. L.; de Castro Pires, V. T.; de Almeida, P. M. L.; Murai, F. “Delator: Money laundering detection via multi-task learning on large transaction graphs”. In: 2022 IEEE International Conference on Big Data (Big Data), 2022, pp. 709–714.

[6] Bodria, F.; Guidotti, R.; Giannotti, F.; Pedreschi, D. “Transparent latent space counterfactual explanations for tabular data”. In: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), 2022, pp. 1–10.

[7] Busson, A.; Rocha, R.; Gaio, R.; Miceli, R.; Pereira, I.; Moraes, D.; Colcher, S.; Veiga, A.; Rizzi, B.; Evangelista, F.; Santos, L.; Marques, F.; Rabaioli, M.; Feldberg, D.; Mattos, D.; Pasqua, J.; Dias, D. “Hierarchical classification of financial transactions through context-fusion of transformer-based embeddings and taxonomy-aware attention layer”. In: Anais do II Brazilian Workshop on Artificial Intelligence in Finance, 2023, pp. 13–24.

[8] Carcillo, F.; Le Borgne, Y.-A.; Caelen, O.; Kessaci, Y.; Oblé, F.; Bontempi, G. “Combining unsupervised and supervised learning in credit card fraud detection”, *Information Sciences*, vol. 557, 2021, pp. 317–331.

[9] Chen, H.; Chen, L. “An application of xgboost algorithm for online transaction fraud detection based on improved sailfish optimizer”. In: 2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2022, pp. 294–299.

[10] Chen, Z.; Khoa, L. D.; Teoh, E. N.; Nazir, A.; Karuppiah, E. K.; Lam, K. S. “Machine learning techniques for anti-money laundering (aml) solutions in suspicious transaction detection: a review”, *Knowl. Inf. Syst.*, vol. 57–2, Nov. 2018, pp. 245–285.

[11] Chen, Z.; Soliman, W. M.; Nazir, A.; Shorfuzzaman, M. “Variational autoencoders and wasserstein generative adversarial networks for improving the anti-money laundering process”, *IEEE Access*, vol. 9, 2021, pp. 83762–83785.

[12] Cook, J.; Ramadas, V. “When to consult precision-recall curves”, *The Stata Journal*, vol. 20–1, 2020, pp. 131–148, https://doi.org/10.1177/1536867X20909693.

[13] Crépey, S.; Lehdili, N.; Madhar, N.; Thomas, M. “Anomaly detection in financial time series by principal component analysis and neural networks”, *Algorithms*, vol. 15–10, 2022.

[14] Dileep, A.; Karthik, A.; Krishna, G. S.; Ganesh, D.; Hariharan, S. “Financial fraud detection using deep learning techniques”. In: 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 2023, pp. 1–6.

[15] Financial Action Task Force. “International Standards on Combating Money Laundering and the Financing of Terrorism & Proliferation: The FATF Recommendations”. Accessed: [Your access date], Source: https://www.fatf-gafi.org/content/dam/fatf-gafi/recommendations/FATF%20Recommendations%202012.pdf, November 2023.

[16] Främling, K. “Contextual importance and utility: A theoretical foundation”. In: AI 2021: Advances in Artificial Intelligence: 34th Australasian Joint Conference, AI 2021, Sydney, NSW, Australia, February 2–4, 2022, Proceedings, 2022, pp. 117–128.

[17] Gangadhar, K. S. N. V. K.; Kumar, B. A.; Vivek, Y.; Ravi, V. “Chaotic variational auto encoder based one class classifier for insurance fraud detection”. 2212.07802, Source: https://arxiv.org/abs/2212.07802, 2022.

[18] Geiger, A.; Liu, D.; Alnegheimish, S.; Cuesta-Infante, A.; Veeramachaneni, K. “Tadgan: Time series anomaly detection using generative adversarial networks”. In: 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 33–43.

[19] Gomes, C.; Jin, Z.; Yang, H. “Insurance fraud detection with unsupervised deep learning”, *Journal of Risk and Insurance*, vol. 88–3, 2021, pp. 591–624, https://onlinelibrary.wiley.com/doi/pdf/10.1111/jori.12359.

[20] Jiang, S.; Dong, R.; Wang, J.; Xia, M. “Credit card fraud detection based on unsupervised attentional anomaly detection network”, *Systems*, vol. 11–6, 2023.

[21] Kennedy, R. K. L.; Salekshahrezaee, Z.; Khoshgoftaar, T. M. “A novel approach for unsupervised learning of highly-imbalanced data”. In: 2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI), 2022, pp. 52–58.

[22] Ketenci, U. G.; Kurt, T.; Önal, S.; Erbił, C.; Aktürkoğlu, S.; İlhan, H. Ş. “A time-frequency based suspicious activity detection for anti-money laundering”, *IEEE Access*, vol. 9, 2021, pp. 59957–59967.

[23] Kingma, D. P.; Welling, M. “Auto-encoding variational bayes”. 1312.6114, Source: https://arxiv.org/abs/1312.6114, 2022.

[24] Kitaev, N.; Łukasz Kaiser; Levskaya, A. “Reformer: The efficient transformer”. 2001.04451, Source: https://arxiv.org/abs/2001.04451, 2020.

[25] Kute, D. V.; Pradhan, B.; Shukla, N.; Alamri, A. “Deep learning and explainable artificial intelligence techniques applied for detecting money laundering–a critical review”, *IEEE Access*, vol. 9, 2021, pp. 82300–82317.

[26] Lacruz, F.; Saniie, J. “Applications of machine learning in fintech credit card fraud detection”. In: 2021 IEEE International Conference on Electro Information Technology (EIT), 2021, pp. 1–6.

[27] Le, P.; Zuidema, W. “Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs”. In: Proceedings of the 1st Workshop on Representation Learning for NLP, Blunsom, P.; Cho, K.; Cohen, S.; Grefenstette, E.; Hermann, K. M.; Rimell, L.; Weston, J.; Yih, S. W.-t. (Editors), 2016, pp. 87–93.

[28] Liu, Y.; Lin, Y.; Xiao, Q.; Hu, G.; Wang, J. “Self-adversarial variational autoencoder with spectral residual for time series anomaly detection”, *Neurocomputing*, vol. 458, 2021, pp. 349–363.

[29] Mizher, M. Z.; Nassif, A. B. “Deep cnn approach for unbalanced credit card fraud detection data”. In: 2023 Advances in Science and Engineering Technology International Conferences (ASET), 2023, pp. 1–7.

[30] Naidoo, V.; Du, S. “A deep learning method for the detection and compensation of outlier events in stock data”, *Electronics*, vol. 11–21, 2022.

[31] Niu, Z.; Yu, K.; Wu, X. “Lstm-based vae-gan for time-series anomaly detection”, *Sensors*, vol. 20–13, 2020.

[32] Nori, H.; Jenkins, S.; Koch, P.; Caruana, R. “Interpretml: A unified framework for machine learning interpretability”. 1909.09223, Source: https://arxiv.org/abs/1909.09223, 2019.

[33] Priscilla, C. V.; Prabha, D. P. “Influence of optimizing xgboost to handle class imbalance in credit card fraud detection”. In: 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 1309–1315.

[34] Reddy, N. M.; Sharada, K. A.; Pilli, D.; Paranthaman, R.; Reddy, K. S.; Chauhan, A. “Cnn-bidirectional lstm based approach for financial fraud detection and prevention system”. In: 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), 2023, pp. 541–546.

[35] Reite, E. J.; Karlsen, J.; Westgaard, E. G. “Improving client risk classification with machine learning to increase anti-money laundering detection efficiency”, *Journal of Money Laundering Control*, vol. ahead-of-print–ahead-of-print, Jan 2024.

[36] Rusanov, G.; Pudovochkin, Y. “Money laundering in the modern crime system”, *Journal of Money Laundering Control*, vol. 24–4, Jan 2021, pp. 860–868.

[37] Shao, M.; Gu, N. “Anomaly detection algorithm based on semi-supervised collaborative strategy”, *Journal of Physics: Conference Series*, vol. 1944–1, jun 2021, pp. 012017.

[38] Sheth, A.; Gaur, M.; Roy, K.; Faldu, K. “Knowledge-intensive language understanding for explainable ai”, *IEEE Internet Computing*, vol. 25, 2021, pp. 19–24.

[39] Shin, A.-H.; Kim, S. T.; Park, G.-M. “Time series anomaly detection using transformer-based gan with two-step masking”, *IEEE Access*, vol. 11, 2023, pp. 74035–74047.

[40] Siblini, W.; Fréry, J.; He-Guelton, L.; Oblé, F.; Wang, Y.-Q. “Master your metrics with calibration”. In: Advances in Intelligent Data Analysis XVIII, Berthold, M. R.; Feelders, A.; Krempl, G. (Editors), 2020, pp. 457–469.

[41] Silva, I. D. G.; Correia, L. H. A.; Maziero, E. G. “Graph neural networks applied to money laundering detection in intelligent information systems”. In: Proceedings of the XIX Brazilian Symposium on Information Systems, 2023, pp. 252–259.

[42] Tallboys, J.; Zhu, Y.; Rajasegarar, S. “Identification of stock market manipulation with deep learning”. In: Advanced Data Mining and Applications, Li, B.; Yue, L.; Jiang, J.; Chen, W.; Li, X.; Long, G.; Fang, F.; Yu, H. (Editors), 2022, pp. 408–420.

[43] Tatulli, M. P.; Paladini, T.; D’Onghia, M.; Carminati, M.; Zanero, S. “Hamlet: A transformer based approach for money laundering detection”. In: Cyber Security, Cryptology, and Machine Learning, Dolev, S.; Gudes, E.; Paillier, P. (Editors), 2023, pp. 234–250.

[44] Tingfei, H.; Guangquan, C.; Kuihua, H. “Using variational auto encoding in credit card fraud detection”, *IEEE Access*, vol. 8, 2020, pp. 149841–149853.

[45] Tuli, S.; Casale, G.; Jennings, N. R. “Tranad: deep transformer networks for anomaly detection in multivariate time series data”, *Proc. VLDB Endow.*, vol. 15–6, Feb. 2022, pp. 1201–1214.

[46] ULB, M. L. G. “Credit card fraud detection”. Source: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, 2017.

[47] Vassallo, D.; Vella, V.; Ellul, J. “Application of gradient boosting algorithms for anti-money laundering in cryptocurrencies”, *SN Computer Science*, vol. 2–3, Mar 2021, pp. 143.

[48] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. “Attention is all you need”. 1706.03762, Source: https://arxiv.org/abs/1706.03762, 2023.

[49] Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; Qi, Y. “A semi-supervised graph attentive network for financial fraud detection”. In: 2019 IEEE International Conference on Data Mining (ICDM), 2019, pp. 598–607.

[50] Wang, X.; Du, Y.; Lin, S.; Cui, P.; Shen, Y.; Yang, Y. “advae: A self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection”, *Knowledge-Based Systems*, vol. 190, 2020, pp. 105187.

[51] Weber, M.; Chen, J.; Suzumura, T.; Pareja, A.; Ma, T.; Kanezashi, H.; Kaler, T.; Leiserson, C. E.; Schardl, T. B. “Scalable graph learning for anti-money laundering: A first look”. 1812.00076, Source: https://arxiv.org/abs/1812.00076, 2018.

[52] Wei, Y.; Qi, Y.; Ma, Q.; Liu, Z.; Shen, C.; Fang, C. “Fraud detection by machine learning”. In: 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2020, pp. 101–115.

[53] Xu, B.; Wang, Y.; Liao, X.; Wang, K. “Efficient fraud detection using deep boosting decision trees”, *Decision Support Systems*, vol. 175, 2023, pp. 114037.

[54] Yang, W.; Wang, R.; Wang, B. “Detection of anomaly stock price based on time series deep learning models”. In: 2020 Management Science Informatization and Economic Innovation Development Conference (MSIEID), 2020, pp. 110–114.

[55] Yokkampon, U.; Mowshowitz, A.; Chumkamon, S.; Hayashi, E. “Robust unsupervised anomaly detection with variational autoencoder in multivariate time series data”, *IEEE Access*, vol. 10, 2022, pp. 57835–57849.

[56] Zamini, M.; Montazer, G. “Credit card fraud detection using autoencoder based clustering”. In: 2018 9th International Symposium on Telecommunications (IST), 2018, pp. 486–491.

[57] Zhou, B.; Liu, S.; Hooi, B.; Cheng, X.; Ye, J. “Beatgan: anomalous rhythm detection using adversarially generated time series”. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 4433–4439.

APPENDIX A – FEATURE ENGINEERING

This appendix provides a comprehensive description of the feature engineering process, including detailed formulas, parameter choices, and implementation details. This information is intended for internal review and is omitted from the public version of this document.

A.1 Feature Engineering

We engineered a diverse set of features targeting specific aspects of trading behavior that may signal potential money laundering. These features can be broadly categorized as follows:

A.1.1 Transaction Volume and Value Metrics

- **Total Transaction Value Sum (Equation A.1):** The total value of all transactions within the time window.

$$\text{Total Value Sum} = \sum_{i=1}^{n} v_i, \tag{A.1}$$

  where v_i is the value of the i-th transaction, and n is the total number of transactions.

- **Total Transaction Value Mean and Standard Deviation (Equations A.2 and A.3):** The average and variability of transaction values.

$$\text{Total Value Mean} = \frac{1}{n}\sum_{i=1}^{n} v_i, \tag{A.2}$$

$$\text{Total Value Std} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(v_i - \text{Total Value Mean})^2}. \tag{A.3}$$

- **Transaction Count (Equation A.4):** The total number of transactions within the time window.

$$\text{Transaction Count} = n. \tag{A.4}$$

A.1.2 Day Trading Indicators

Frequent day trading, especially when consistently profitable, can be indicative of suspicious activity. All day trading metrics are calculated over a *fixed window size of 15 days*.

- **Day Trading Result (Equation A.5):** The net profit/loss from day trading activities, calculated as:

$$\text{Day Trading Result} = \sum_{j=1}^{m} q_j \times (p_{\text{sell},j} - p_{\text{buy},j}), \tag{A.5}$$

  where m is the number of day trades, q_j is the minimum quantity traded in the j-th day trade (between buys and sells), and $p_{\text{buy},j}$ and $p_{\text{sell},j}$ are the weighted average purchase and sale prices for that trade.

- **Trading Frequency (Equation A.6):** The proportion of days with day trading activity within the 15-day window:

$$\text{Trading Frequency} = \frac{d_{\text{dt}}}{15}, \tag{A.6}$$

  where d_{dt} is the number of days with day trading activity.

- **Weighted Day-Trade Accuracy Index (Equation A.7):** Combines the success rate of profitable day trades with their frequency:

$$\text{Weighted Day-Trade Accuracy Index} = \left(\frac{d_{\text{positive}}}{d_{\text{dt}}}\right) \times \text{Trading Frequency}, \tag{A.7}$$

  where d_{positive} is the number of days with positive day trading results.

- **Has Day Trade:** A binary indicator (1 if day trading occurred, 0 otherwise) within the 15-day window. This is derived from the *Day Trading Result*.

A.1.3 Counterparty Analysis

Analyzing client-counterparty relationships can reveal potentially suspicious patterns.

- **Concentration of Counterparties (Equation A.8):** The proportion of a client's trades conducted with a specific counterparty c:

$$\text{Concentration of Counterparty}_c = \frac{t_c}{t_{\text{client}}}, \tag{A.8}$$

where t_c is the number of trades with counterparty c, and t_{client} is the client's total number of trades.

- **Maximum and Average Concentration (Equations A.9 and A.10):** Summary metrics of counterparty concentration:

$$\text{Max Concentration} = \max_c \left(\text{Concentration of Counterparty}_c\right), \tag{A.9}$$

$$\text{Avg Concentration} = \frac{1}{k}\sum_{c=1}^{k} \text{Concentration of Counterparty}_c, \tag{A.10}$$

where k is the number of unique counterparties the client interacted with.

- **Number of Counterparties (k):** The total number of unique counterparties for a given client within the analysis window.
- **Number of High Concentration Counterparties:** The number of unique counterparties where the *Concentration of Counterparty*$_c$ exceeds a predefined threshold (50%).
- **Has High Concentration:** A binary indicator flagging if the *Max Concentration* exceeds a threshold (80%).

A.1.4 Transaction Pattern Indicators

These features aim to capture unusual deviations in trading patterns.

- **Amount Variability (Equation A.11):** The coefficient of variation of transaction values, measuring the relative variability of transaction amounts:

$$\text{Amount Variability} = \frac{\text{Total Value Std}}{\text{Total Value Mean}}. \tag{A.11}$$

- **High-Value Transactions (Equation A.12):** The number of transactions exceeding a threshold, defined as the *95th percentile of transaction values* across all clients:

$$\text{High-Value Transactions} = \sum_{i=1}^{n} \mathbb{I}\left(v_i > v_{95\text{th percentile}}\right). \tag{A.12}$$

- **Large Orders (Equation A.13):** The number of transactions where the order quantity (q_i) is significantly larger than the client's usual trading behavior, defined as two

standard deviations above the mean:

$$\text{Large Orders} = \sum_{i=1}^{n} \mathbb{I}\left(q_i > \bar{q} + 2\sigma_q\right), \tag{A.13}$$

where $\bar{q}$ is the client's average order quantity, and σ_q is the standard deviation of the client's order quantities.

- **Frequent Orders (Equation A.14):** The number of transactions with order quantities above the *90th percentile* of the client's order quantity distribution:

$$\text{Frequent Orders} = \sum_{i=1}^{n} \mathbb{I}\left(q_i > Q_{90}\right), \tag{A.14}$$

where Q_{90} represents the 90th percentile.

- **Cross-Market Trades (Equation A.15):** The number of unique markets (m) in which the client has traded within the time window:

$$\text{Cross-Market Trades} = m. \tag{A.15}$$

- **Short Holding Periods (Equation A.16):** A binary indicator (1 if true, 0 otherwise) where a short holding period is defined as a duration less than one hour ($\Delta t = 1$ hour) between the first and last trade within the time window.

$$\text{Short Holding Periods} = \mathbb{I}\left(t_{\max} - t_{\min} < 1 \text{ hour}\right). \tag{A.16}$$

- **Large Price Changes (Equation A.17):** The relative change in price for a given asset, calculated using the maximum ($p_{\max}$) and minimum ($p_{\min}$) prices observed within the time window:

$$\text{Large Price Changes} = \frac{p_{\max} - p_{\min}}{p_{\min}}, \tag{A.17}$$

- **Has High Volume:** A binary indicator (1 if true, 0 otherwise) for whether the *Total Value Sum* for a client is above the 95th percentile of all client's *Total Value Sum* within the time window.

- **Has Frequent Trading:** A binary indicator (1 if true, 0 otherwise) for whether the client's *Transaction Count* is above the 75th percentile of all client's *Transaction Count* within the time window.

A.1.5 Risk Factors

These features incorporate client-specific risk factors that can be used to assess the risk of money laundering activities.

- **Client Age (Idade):** The age of the client, calculated at the time of each transaction. This factor can be used to identify clients in high-risk age demographics.
- **Border City (cidade_fronteira):** A binary indicator (1 if the client is located in a border city, 0 otherwise). Clients in border cities may be considered higher risk due to increased opportunities for cross-border financial crimes.
- **Financial Capacity Mismatch (Equation A.18):** A binary indicator (1 if true, 0 otherwise) flagged when the *Total Value Sum* of transactions significantly exceeds the client's declared financial capacity, using a multiplier (α) of 1.5:

$$\text{Financial Capacity Mismatch} = \begin{cases} 1, & \text{if Total Value Sum} > 1.5 \times \text{Declared Capacity}, \\ 0, & \text{otherwise.} \end{cases} \tag{A.18}$$

  where *Declared Capacity* refers to the client's *Net Worth (Pat. Líquido)* obtained from external financial data. Large discrepancies between transaction volume and declared net worth can indicate suspicious activity.

A.1.6 Unusual Activity Indicators

These features highlight potentially suspicious deviations from typical trading behaviors.

- **After-Hours Trading (Equation A.19):** The number of trades executed outside of regular trading hours (9 am to 5 pm):

$$\text{After-Hours Trading} = \sum_{i=1}^{n} \mathbb{I}\left(h_i < 9 \text{ or } h_i > 17\right), \tag{A.19}$$

  where h_i is the hour of the i-th transaction.

- **Asset Concentration (Equation A.20):** The ratio of unique assets traded (a) to the total number of transactions:

$$\text{Asset Concentration} = \frac{a}{\text{Transaction Count}}. \tag{A.20}$$

- **Circular Trading (Equation A.21):** The number of consecutive trades with the same counterparty:

$$\text{Circular Trading} = \sum_{i=2}^{n} \mathbb{I}\left(c_i = c_{i-1}\right), \tag{A.21}$$

where c_i represents the counterparty in the i-th transaction.

- **Unusual Trading Volume (Equation A.22):** The number of trades with volumes exceeding β times the client's average trading volume ($\bar{v}_{\text{client}}$), where $\beta = 2$:

$$\text{Unusual Trading Volume} = \sum_{i=1}^{n} \mathbb{I}\left(v_i > \beta \times \bar{v}_{\text{client}}\right), \tag{A.22}$$

where v_i is the volume of the i-th transaction and $\bar{v}_{\text{client}}$ represents the client's average trading volume. This average is computed as the mean value of the client's transactions.

- **Rapid Succession Transactions (Equation A.23):** The number of transactions occurring within a short time frame of 10 seconds ($\delta t = 10$ seconds):

$$\text{Rapid Succession Transactions} = \sum_{i=2}^{n} \mathbb{I}\left(t_i - t_{i-1} < 10 \text{ seconds}\right), \tag{A.23}$$

where t_i represents the timestamp of the i-th transaction.

A.1.7 Additional Alerts

These features are designed to detect specific transaction patterns that could be associated with money laundering attempts.

- **Structuring Pattern (Equation A.24):** The number of transactions falling within a specific range below certain thresholds, between 9,000 and 10,000 currency units:

$$\text{Structuring Pattern} = \sum_{i=1}^{n} \mathbb{I}\left(9000 < v_i < 10000\right), \tag{A.24}$$

where v_i is the transaction value.

- **Round Number Transactions (Equation A.25):** The number of transactions with values that are multiples of 1,000 currency units ($\gamma = 1000$):

$$\text{Round Number Transactions} = \sum_{i=1}^{n} \mathbb{I}\left(v_i \bmod 1000 = 0\right), \tag{A.25}$$

- **Counterparty Diversity (Equation A.26):** A measure of the distribution of trades across different counterparties:

$$\text{Counterparty Diversity} = 1 - \sum_{c=1}^{k} \left(\frac{t_c}{t_{\text{client}}} \right)^2 . \tag{A.26}$$

- **Counterparty Frequency (Equation A.27):** The ratio of unique counterparties to the total number of transactions:

$$\text{Counterparty Frequency} = \frac{\text{Unique Counterparties}}{\text{Transaction Count}}. \tag{A.27}$$

- **Transaction Time Variance (Equation A.28):** The variance in the hour of day (h_i) of transactions:

$$\text{Transaction Time Variance} = \text{Var}\left(h_i\right). \tag{A.28}$$

- **Asset Turnover Rate (Equation A.29):** The ratio of the total transaction value to the client's net worth:

$$\text{Asset Turnover Rate} = \frac{\text{Total Value Sum}}{\text{Net Worth}}. \tag{A.29}$$

- **Average Price Deviation (Equation A.30):** The average deviation of the transaction price (p_i) from the average market price ($\bar{p}_{\text{market},i}$) at the time of the transaction:

$$\text{Average Price Deviation} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{p_i - \bar{p}_{\text{market},i}}{\bar{p}_{\text{market},i}} \right| . \tag{A.30}$$

- **Transaction Size Consistency (Equation A.31):** The coefficient of variation of transaction values, measuring the consistency of transaction sizes:

$$\text{Transaction Size Consistency} = \frac{\sigma_v}{\bar{v}}, \tag{A.31}$$

  where σ_v is the standard deviation of transaction values, and $\bar{v}$ is the mean transaction value.

- **Unusual Hour Trading Percentage (Equation A.32):** The percentage of trades occurring outside of typical trading hours (9 am to 5 pm):

$$\text{Unusual Hour Trading Percentage} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left(h_i < 9 \text{ or } h_i > 17\right). \tag{A.32}$$

A.1.8 Swing Trading

- **Swing Trading:** A trading strategy characterized by holding positions for longer than one day, aiming to profit from price swings or trends.
- **Number of Swing Trades:** The number of trades identified as swing trades within the time window, which have a holding period exceeding one day.
- **Average Holding Period:** The average length of time (in days) for which a swing trade position is held.

A.2 Data Preprocessing

A.2.1 Data Preprocessing and Imputation

Prior to analysis, the dataset was preprocessed to ensure data quality and suitability for anomaly detection. This involved removing duplicate records and features deemed irrelevant for this task. Missing values were handled through imputation strategies chosen based on the nature of each feature:

- **Numerical Features:** Numerical features were imputed using either zero or the median value. Zero imputation was applied to features representing financial sums or counts, where a missing value logically indicates a zero quantity. The median was used to impute missing values in other numerical features, providing a robust measure of central tendency.
- **Categorical Features:** For categorical features, missing values were imputed with the mode, representing the most frequently observed category within each feature.

A.2.2 Feature Scaling and Normalization

All continuous features were scaled to the range $[0, 1]$ using min-max normalization:

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{A.33}$$

Binary features were converted to floating-point representations (0.0 and 1.0) to ensure data uniformity. Infinity values resulting from calculations, such as division by zero, were replaced with the median value of the feature.

A.2.3 Correlation Analysis

To address potential multicollinearity, we conducted a pairwise correlation analysis across all features. We established a threshold of 0.9 for the absolute Pearson correlation coefficient, classifying feature pairs exceeding this value as highly correlated. This analysis led to the removal of three features—Avg Concentration Counterpart, Has High Concentration, and Amount Variability—due to their high correlation with other variables in the dataset.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br