

# **Comparative Analysis of Supervised MLP and Unsupervised Autoencoder for Credit Card Fraud Detection**

## **Abstract**

This research addresses the critical challenge of detecting fraudulent transactions in the financial sector, characterized by extreme class imbalance (0.17%). We utilized the Kaggle Credit Card Fraud Detection dataset to evaluate two deep learning architectures: a supervised Multi-Layer Perceptron (MLP) and an unsupervised Autoencoder. To maximize model performance, we implemented automated hyperparameter optimization using Optuna framework, tuning architecture depth, learning rates, and loss functions. Results indicate that the supervised MLP, enhanced with Batch Normalization and Class Weighting, achieved a superior Area Under the Precision-Recall Curve (AUPRC) of 0.74. The Autoencoder, optimized for anomaly detection with a compressed bottleneck, showed a lower AUPRC of 0.17, highlighting the difficulty of unsupervised separation in high-dimensional feature spaces, yet demonstrated potential for detecting anomalies at specific calibrated thresholds.

*Keywords: Fraud Detection, Deep Learning, Autoencoder, MLP, Class Imbalance*

## **A. Introduction**

The exponential growth of digital financial transactions has led to a parallel increase in fraudulent activities, posing significant risks to global banking systems. Traditional rule-based systems often fail to adapt to complex, non-linear fraud patterns, necessitating the adoption of advanced Machine Learning (ML) techniques. The primary objective of this research is to evaluate the effectiveness of Deep Learning models in identifying fraudulent credit card transactions. Specifically, we compare a supervised Multi-Layer Perceptron (MLP), which relies on labeled data, against an unsupervised Autoencoder, which detects

fraud as statistical anomalies. This comparison is vital for understanding the trade-off between precision and recall in real-world scenarios where fraudulent examples are scarce.

## **B. Literature Review**

In light of the escalating volume of digital financial transactions, numerous studies have explored innovative solutions leveraging artificial intelligence to enhance fraud detection systems. The challenge of identifying fraudulent activities in real-time requires robust and adaptive models. [1] Jain et al. (2024) addressed the challenge of securing credit card systems using classical algorithms. They developed “FraudFort,” harnessing Support Vector Machines and Random Forests. While Random Forest achieved high accuracy, the authors noted limitations in computational cost when scaling to high-frequency streams. Complementing this, [2] Mahesh et al. (2025) focused on banking fraud detection, emphasizing feature extraction to handle severe class imbalance. Their work highlights that refining input features significantly boosts performance, a principle we adopt in our preprocessing stage.

Addressing the need for autonomy, [3] Singh et al. (2024) investigated real-time detection using Autoencoders. Unlike supervised models, this approach utilizes unsupervised learning to model normal behavior and flag deviations. Their results demonstrated high effectiveness for zero-day attacks, directly motivating our choice of the Autoencoder as a primary model. Furthermore, [4] Xu et al. (2025) and [5] Abid et al. (2025) proposed hybrid architectures combining supervised and unsupervised learning to mitigate false positives and balance the accuracy-latency trade-off. These studies collectively suggest that the future of fraud detection lies in architectures that can handle imbalance and generalize well, validating our comparative approach.

### **C. Data**

We utilized the "Credit Card Fraud Detection" dataset sourced from Kaggle (ULB Machine Learning Group). The dataset contains 284,807 transactions made by European cardholders in September 2013. The dataset includes 30 features: Time, Amount, and 28 principal components (V1–V28) obtained via PCA transformation to protect user privacy. The dataset is highly imbalanced, with only 492 fraudulent transactions (0.172%) compared to 284,315 legitimate ones. Unlike traditional approaches that scale only specific columns, we applied MinMaxScaler to all 30 features, mapping them to the range. This step was critical for the stability of the Autoencoder, particularly when using Sigmoid activation functions in the output layer. A strict time-based split was implemented (First 80% for Training, last 20% for Testing) to prevent data leakage. Additionally, a validation set was isolated from the training data to monitor convergence and perform Early Stopping.

### **D. Methodology**

We implemented and compared two distinct neural network architectures using Python and PyTorch. Instead of a simple feed-forward network, we designed an Advanced MLP incorporating: Batch Normalization after each linear layer to stabilize learning. LeakyReLU activation (slope 0.2) to prevent the "dying ReLU" problem. We used BCEWithLogitsLoss with a calculated `pos_weight` to heavily penalize missing fraud cases (False Negatives). Optuna was used to search for the optimal number of layers (2–4), hidden units (32–256), learning rates (1e-4, 1e-2) and Dropout rates (0.1–0.5).

We treated fraud detection as an anomaly detection task. The model was trained *only* on legitimate transactions (Class 0) to learn the latent representation of "normal" behavior.

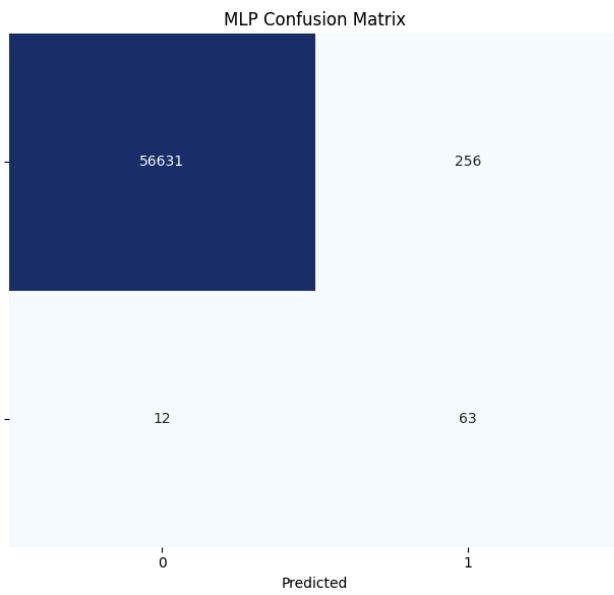
The size of the latent space (bottleneck) is the most critical parameter. We used Optuna to dynamically select the bottleneck size (between 2 and 8 neurons). A tighter bottleneck forces the model to learn only the most robust features, preventing it from "memorizing" noise. We evaluated multiple reconstruction loss functions, including Mean Squared Error (MSE), Mean Absolute Error (L1), and Binary Cross-Entropy (BCE). The optimization process selected the loss function that maximized the validation AUPRC. Both models were trained using the AdamW optimizer with a ReduceLROnPlateau scheduler. To prevent overfitting, we implemented Early Stopping with a patience of 6–10 epochs, monitoring the validation loss.

## E. Experiments & Results

Given the extreme class imbalance, standard Accuracy is misleading. We prioritized the Area Under the Precision-Recall Curve (AUPRC), which summarizes the trade-off between Precision and Recall across all possible thresholds.

Model	AUPRC	Precision	Recall	F1-Score
MLP (Supervised)	0.7449	0.1975	0.8400	0.3198
Autoencoder	0.1689	0.0662	0.7733	0.1220

*Table 1: Performance Comparison for Fraud Class*



0 for non-fraud and 1 for fraud classes

Figure 1: MLP Confusion Matrix

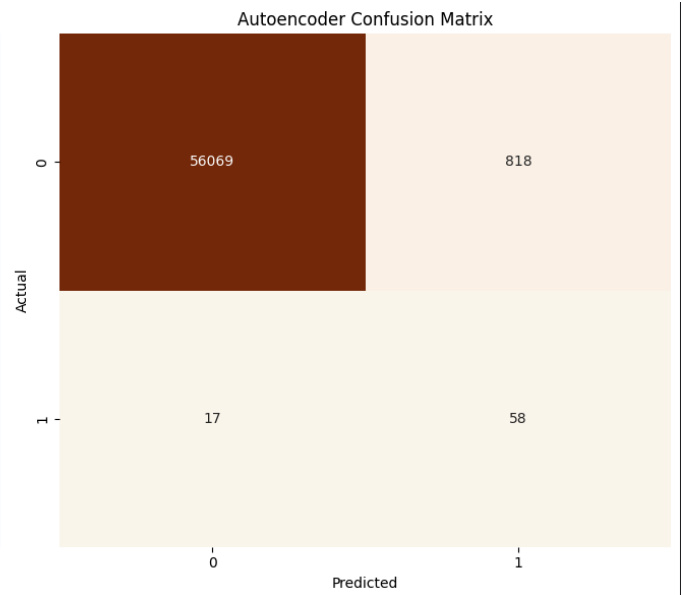


Figure 2: Autoencoder Confusion Matrix

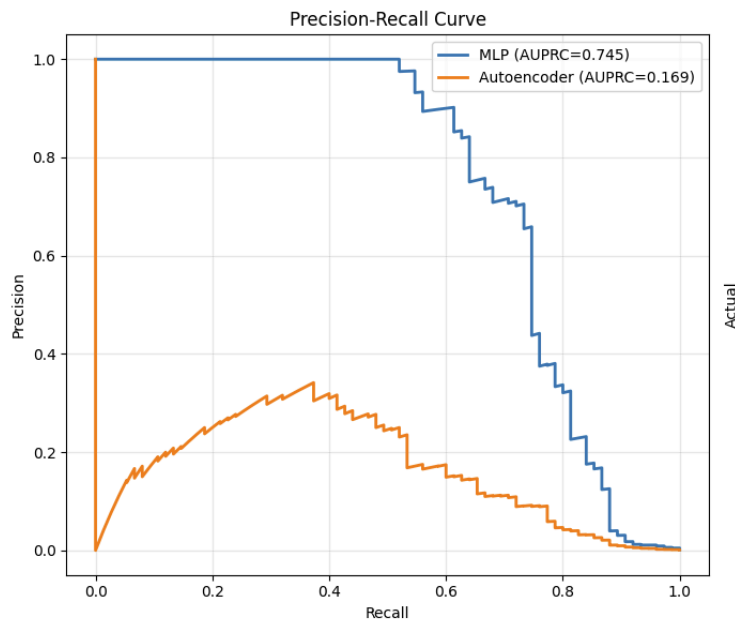


Figure 3: Precision-Recall Curve for Fraud class

The supervised MLP demonstrated significantly higher stability (AUPRC 0.74) (Figure 3). By explicitly learning from labeled fraud examples, it established a robust decision boundary, maintaining high precision even at high recall rates. While the Autoencoder achieved a lower AUPRC (0.17) (Figure 3), the Confusion Matrix analysis at

the calculated threshold (Mean + 3 Std Dev) showed it successfully identified 58 out of 75 fraud cases (*Figure 2*). However, the low Precision (0.07) (*Table 1*) indicates a high number of False Positives. This suggests that while the Autoencoder successfully flags fraud as "anomalous," there are many legitimate transactions that are statistically similar to fraud in the latent space (high reconstruction error). The Autoencoder is effective as a "first-pass" filter but requires a secondary verification stage.

## **F. Conclusion**

This study compared supervised and unsupervised deep learning approaches for fraud detection. Through automated hyperparameter tuning (Optuna), we demonstrated that a supervised Advanced MLP significantly outperforms an unsupervised Autoencoder in terms of AUPRC (0.74 vs 0.17) (*Figure 3*) on historical data. However, the Autoencoder remains valuable for its ability to detect anomalies without relying on labeled historical fraud patterns. Future work should focus on a Hybrid Ensemble, where the Autoencoder's reconstruction error is used as an additional feature for the MLP, potentially combining the strengths of both approaches.

## **References**

- [1] S. Jain, N. Sharma, and M. Kumar, "Fraudfort: Harnessing machine learning for credit card fraud detection," in 2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP), 2024, pp. 41–46.
- [2] P. Mahesh, S. C. M, P. M, J. K. C, A. R, and K. G, "Credit card fraud detection in banking using machine learning," in 2025 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), 2025, pp. 1–7.

- [3] M. Singh, N. Bansal, G. S. K. Kavithamani, M. Almusawi, and C. P. Patnaik, “Real-time fraud detection in financial transactions using autoencoders,” in 2024 International Conference on Advances in Computing, Communication and Materials (ICACCM), 2024, pp. 1–4.
- [4] S. Xu, Y. Cao, Z. Wang, and Y. Tian, “Fraud detection in online transactions: Toward hybrid supervised–unsupervised learning pipelines,” in 2025 6th International Conference on Electronic Communication and Artificial Intelligence (ICECAI), 2025, pp. 470–474.
- [5] S. B. Abid, B. Fuchs, G. Haberkorn, K. Jepsen, A. Krampetz, and D. Matthes, “A scalable hybrid approach to detecting fraud with machine learning,” in 2025 33rd European Signal Processing Conference (EUSIPCO), 2025, pp. 1297–1301.