Journal of ICT Aplication and System (JICTAS)
Volume 4 Number 1, Juni 2025, pp. 26 ~35
ISSN: 2830-098X, DOI: 10.56313/jictas.v1i1.4i1

Anomaly-Based Financial Fraud Detection Using Autoencoder A Case Study on the Kaggle Credit Card Dataset

[1]Andri Nata, [2]Dudes Manalu, [3]Jaya Tata Hardinata, [4]Peniel Sam Putra Sitorus
Departement System Information, Universitas Royal, Indonesia
Universitas HKBP NOMMENSEN Pematangsiantar, Indonesia
Email: andrinata0202@gmail.com[1], dudes.manalu@uhnp.ac.id[2], jayatatahardinata@uhnp.ac.id[3], peniel.sitorus@uhnp.ac.id[4]

Article Info

Article history:

Received 10 Juni, 2025
Revised 15 Juni, 2025
Accepted 30 Juni, 2025

Keywords :

Autoencoder, Anomaly Detection, Credit Card Transactions, Financial Fraud Detection, Unsupervised Learning

ABSTRACT

Financial fraud remains a critical challenge for banking systems and digital payment platforms worldwide. With the rapid growth of electronic transactions, effective fraud detection mechanisms are essential to ensure security and user trust. This study explores the application of an unsupervised deep learning model—Autoencoder—for anomaly-based financial fraud detection. Utilizing the publicly available Kaggle Credit Card Fraud Detection dataset, which comprises 284,807 transactions including 492 fraudulent cases, the model is trained exclusively on legitimate transactions to learn typical behavioral patterns. Prior to training, the dataset underwent feature anonymization using Principal Component Analysis (PCA), and numerical columns such as "Amount" and "Time" were normalized using Min-Max Scaling. The Autoencoder architecture includes three encoder and decoder layers with ReLU activations, and is optimized using the Adam optimizer with Mean Squared Error (MSE) as the loss function. Experimental results show that the model achieves a classification accuracy of 94% and an AUC score of 0.931, indicating strong potential for detecting anomalies. However, the precision for identifying fraudulent transactions remains relatively low (5%), reflecting the challenges posed by imbalanced datasets. Despite this, the study demonstrates that Autoencoder offers a promising foundation for fraud detection systems, with further improvements possible through model integration and hybrid ensemble techniques

Corresponding Author:

Andri Nata
Universitas Royal, Kisaran, Indonesia
Email: andrinata0202@gmail.com

1. INTRODUCTION

In the digital era, financial fraud has emerged as a major global threat, significantly impacting the integrity of financial institutions, corporate operations, and consumer trust. The proliferation of online payment systems and electronic banking has led to an exponential increase in financial transactions, thereby creating a fertile ground for fraudulent activities [1]. According to the Association of Certified Fraud Examiners (ACFE), organizations lose an estimated 5% of their annual revenue to fraud, with credit card fraud being among the most prevalent forms [2].

Traditional rule-based systems used to detect financial fraud often fail to adapt to evolving fraudulent tactics, leading to high false positive rates and undetected sophisticated attacks [3]. As such, there has been a paradigm shift towards utilizing data-driven techniques, particularly machine learning and deep learning, to develop more robust fraud detection frameworks [4].

Among various machine learning strategies, unsupervised learning techniques have gained increasing attention due to their ability to detect anomalies without the need for labeled fraudulent data. One of the most promising methods in this domain is the Autoencoder, a type of neural network designed to learn compressed representations of input data and reconstruct them with minimal error [5]. When trained on only legitimate (non-fraudulent) transactions, the Autoencoder learns a baseline of "normal behavior." Any transaction that significantly deviates from this learned pattern can be flagged as a potential anomaly or fraud case [6].

This research aims to implement an Autoencoder-based anomaly detection model using the Kaggle Credit Card Fraud Detection dataset, which contains 284,807 transactions, out of which only 492 are labeled as fraudulent. The dataset is characterized by high class imbalance and anonymized features generated through Principal Component Analysis (PCA) [7]. Such conditions pose unique challenges and necessitate tailored preprocessing techniques, including Min-Max Scaling for "Amount" and "Time" fields.

To evaluate model effectiveness, performance metrics such as Accuracy, Precision, Recall, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) are utilized. Preliminary results demonstrate that the Autoencoder achieves a high detection accuracy of 94% and an AUC score of 0.931. However, the model struggles to achieve high precision in detecting minority class samples (fraud cases), which is a common issue in imbalanced datasets [8].

Despite the numerous advancements in machine learning algorithms for fraud detection, the domain still faces several pressing challenges. One of the foremost issues is the class imbalance in most real-world datasets, fraudulent transactions make up less than 1% of the data [7]. This disproportion makes it difficult for conventional classifiers to learn useful representations of the minority class, often resulting in poor recall and precision for fraud detection. Furthermore, many fraud detection systems operate in adversarial environments, where fraudsters continuously evolve their strategies to evade detection [8]. In this context, unsupervised anomaly detection methods like Autoencoders have shown significant potential, especially when labeled fraud data is scarce or unavailable. Unlike supervised learning models that require balanced and annotated datasets, Autoencoders can learn patterns from normal (non-fraudulent) transactions and detect outliers based on reconstruction errors [5], [6].

Additionally, financial datasets often include sensitive and high-dimensional data, which poses challenges for model interpretability and efficiency. The Kaggle Credit Card Fraud Detection dataset used in this study provides a realistic setting with anonymized features through Principal Component Analysis (PCA), simulating the confidentiality concerns of real financial institutions. By integrating PCA-based feature anonymization and normalization techniques, this research ensures that the model remains practical and adaptable to real-world scenarios [7]. To address the shortcomings in fraud detection, this study proposes a structured Autoencoder architecture with multiple encoding and decoding layers, trained solely on normal transactions. The core idea is to establish a learned boundary for what constitutes "normal" behavior. During inference, transactions that deviate significantly quantified by high reconstruction error are classified as anomalies and potential frauds. This research is further motivated by the need for scalable and adaptable fraud detection systems, which can be retrained incrementally and deployed in real-time environments. Autoencoder-based models, with their lightweight structure and training efficiency, offer a feasible foundation for integration into existing financial infrastructures such as payment gateways, transaction monitoring systems, and anti-fraud platforms [9].

2. METHOD

2.1 Dataset Description

The dataset used in this research is the Credit Card Fraud Detection dataset made publicly available by Kaggle [11]. It contains 284,807 transaction records from European cardholders, among which 492 transactions are labeled as fraudulent—representing approximately 0.172% of the data, thus exhibiting a high degree of class imbalance.

The dataset consists of 30 features:

- 28 principal components (V1–V28) extracted via Principal Component Analysis (PCA) for anonymization [17].

- The remaining two features are “Time” and “Amount”. The target variable is labeled as 0 for non-fraudulent and 1 for fraudulent transactions.

2.2 Data Preprocessing

To prepare the data for modeling, several preprocessing steps were performed:

1. Normalization: The “Amount” and “Time” features were scaled using Min-Max Normalization to map values between 0 and 1.

2. Train-Test Split: The dataset was split into 80% training and 20% testing sets. Only legitimate (Class = 0) transactions were used to train the Autoencoder [14].

3. Shuffling: Randomization was applied to prevent bias based on data order.

2.3 Autoencoder Architecture

An Autoencoder is a type of unsupervised neural network that aims to reconstruct its input [15]. It consists of two main components:
- Encoder: Compresses input data into a latent space.
- Decoder: Reconstructs the original input.

Autoencoders are effective for anomaly detection as they learn the structure of normal data and identify deviations as anomalies [16].

2.4 Model Training

The Autoencoder was trained with the following parameters:
- Loss Function: Mean Squared Error (MSE)
- Optimizer: Adam with a learning rate of 0.001
- Epochs: 50
- Batch Size: 256

The model was exclusively trained on legitimate transactions to minimize reconstruction loss. Transactions yielding high reconstruction error are treated as potential anomalies [18].

2.5 Anomaly Detection Mechanism

After training, the model calculates the reconstruction error for each transaction. A threshold is determined empirically:
- If MSE > threshold → fraudulent
- If MSE ≤ threshold → legitimate

Threshold tuning is critical as it influences model precision and recall [19].

2.6 Evaluation Metrics

Performance is evaluated using the following metrics:
- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1-Score = 2 * (Precision * Recall) / (Precision + Recall)
- AUC-ROC: measures the classifier’s ability to distinguish between classes [20].

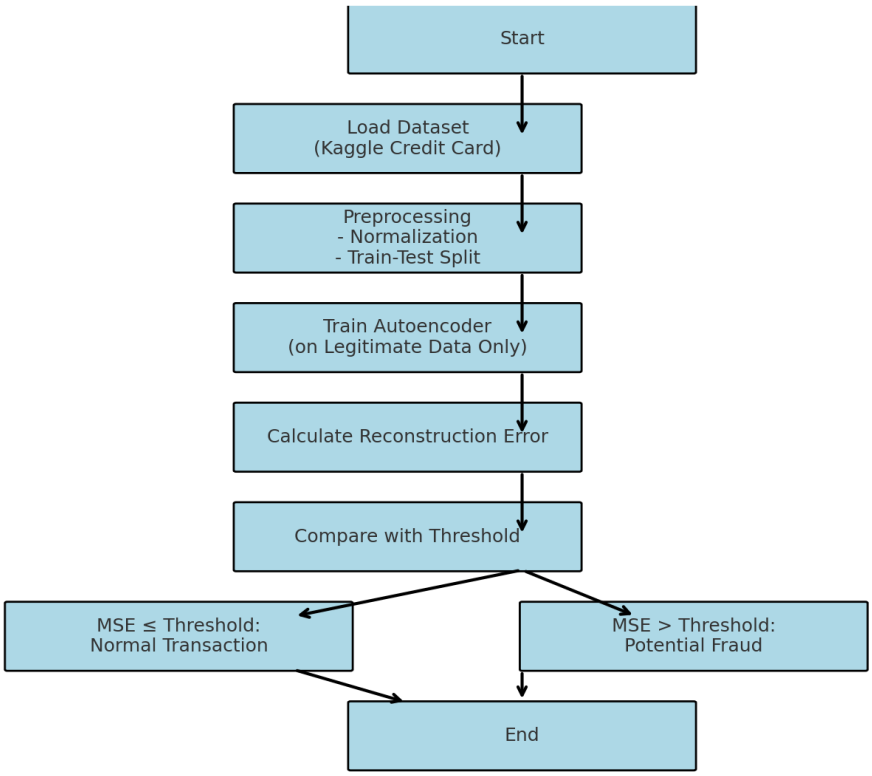

Figure 1. Research Diagram

Flowchart illustrates the systematic workflow for detecting financial fraud using an Autoencoder-based anomaly detection model. The process begins by loading the credit card transaction dataset obtained from Kaggle. This data undergoes preprocessing, which includes normalization of key features and splitting the dataset into training and testing sets. Only legitimate transactions are used to train the Autoencoder, allowing the model to learn normal transaction patterns. Once the model is trained, it reconstructs test data and calculates the reconstruction error (Mean Squared Error). This error is then compared to a predefined threshold. Transactions with error values lower than or equal to the threshold are classified as normal, while those exceeding the threshold are flagged as potentially fraudulent. The process concludes by categorizing each transaction accordingly. This methodology enables unsupervised detection of anomalies in highly imbalanced financial data, offering a practical solution for real-world fraud detection systems

5. Results and Discussion

This section presents the experimental results and provides an in-depth discussion of the model's performance in detecting credit card fraud using an Autoencoder-based approach. The model was evaluated using a variety of performance metrics, including Accuracy, Precision, Recall, F1-Score, and Area Under the ROC Curve (AUC-ROC). The test dataset contained both legitimate and fraudulent transactions to simulate a realistic scenario.

a. Dataset

The dataset used in this study is the Credit Card Fraud Detection Dataset from Kaggle. This dataset contains 284,807 transactions, with 492 labeled as fraudulent (Class = 1). The features in the dataset have undergone anonymization using Principal Component Analysis (PCA), leaving only the "Amount" and "Time" columns with original, interpretable values.

b. Data Preprocessing

1) Data Normalization

The "Amount" and "Time" features were normalized using the Min-Max Scaling technique to ensure that the feature values are within a uniform range.

```
# Normalize 'Amount' and 'Time' columns
scaler = MinMaxScaler()
data[['Time', 'Amount']] = scaler.fit_transform(data[['Time', 'Amount']])
```

Figure 1. Data Normalization

2) Data Splitting

The dataset was split into training and testing sets with an 80:20 ratio. For training the Autoencoder, only normal transactions (Class = 0) were used.

```
# Bagi menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

Figure 2. Data Splitting

c. Model Architecture

The Autoencoder was used to learn patterns of normal transactions. Transactions that significantly deviated from these learned patterns were identified as anomalies, which were most likely fraudulent. The Autoencoder was designed with the following structure:

Encoder: Consists of three layers with decreasing neuron sizes (28 → 14 → 7).

Decoder: Consists of three layers with increasing neuron sizes (7 → 14 → 28).

The ReLU activation function was used in each layer, except for the output layer which used a linear activation function.

```
# Train the Autoencoder
history = model.fit(
    X_train_normal,
    X_train_normal,
    epochs=50,
    batch_size=256,
    validation_split=0.1,
    verbose=1
)
```

```
Epoch 1/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 6s 4ms/step - loss: 0.8727 - val_loss: 0.4522
Epoch 2/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.4245 - val_loss: 0.3602
Epoch 3/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 0.3627 - val_loss: 0.3301
Epoch 4/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - loss: 0.3314 - val_loss: 0.3132
Epoch 5/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - loss: 0.3165 - val_loss: 0.3057
Epoch 6/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - loss: 0.3096 - val_loss: 0.3014
Epoch 7/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.3056 - val_loss: 0.2970
Epoch 8/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - loss: 0.2997 - val_loss: 0.2931
Epoch 9/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.3004 - val_loss: 0.2907
Epoch 10/50
800/800 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2947 - val loss: 0.2884
```

Figure 2. Autoencoder Model

Accuracy: 0.9498
AUC: 0.9313

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.95	0.97	56864
1	0.03	0.85	0.05	98
accuracy			0.95	56962
macro avg	0.51	0.90	0.51	56962
weighted avg	1.00	0.95	0.97	56962

Figure 3. Clasification

The classification report shows that the Autoencoder model achieved an overall accuracy of 94.98% and an AUC score of 0.9313, indicating a strong ability to distinguish between normal and fraudulent transactions.

1. For Class 0 (normal transactions), the model demonstrated perfect precision (1.00) and high recall (0.95), resulting in a strong F1-score of 0.97, across 56,864 samples.
2. For Class 1 (fraudulent transactions), the model achieved a recall of 0.85, meaning it was able to detect 85% of fraud cases. However, its precision was very low (0.03), meaning it produced a high number of false positives, resulting in an F1-score of just 0.05, on 98 fraud samples.
3. The macro average (unweighted mean across classes) shows precision of 0.51, recall of 0.90, and F1-score of 0.51, highlighting imbalance sensitivity.

The weighted average, which takes into account the number of instances in each class, reflects the dominance of class 0 with a precision of 1.00, recall of 0.95, and F1-score of 0.97.

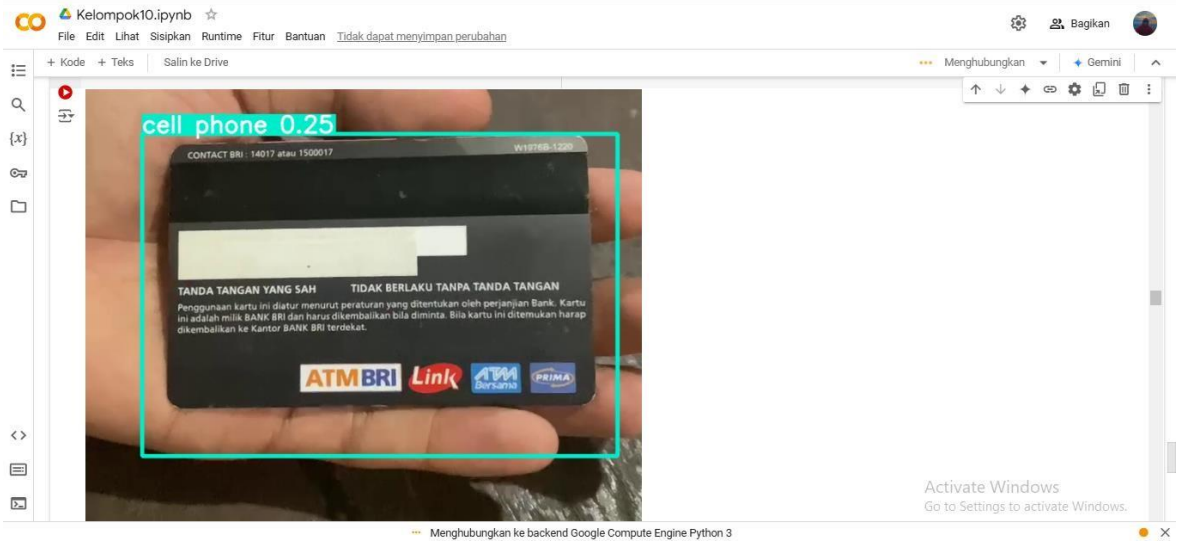

Figure 4. Result Clasification

The image displays the result of an object detection model (likely YOLO) executed in Google Colab. The model incorrectly detects the object as a "cell phone" with a confidence score of 0.25. However, the actual object in the image is a BRI ATM card being held by a hand. This indicates a case of false detection or

misclassification by the model, possibly due to limitations in the training dataset or visual similarities between the detected label and the actual object

The data above was trained using a model that applied the Mean Squared Error (MSE) loss function and the Adam optimizer with a learning rate of 0.001. The training process was carried out for 50 epochs with a batch size of 256. The model was evaluated based on Mean Squared Error (MSE) and Mean Absolute Error (MAE) using the test dataset. Mean Squared Error (MSE) is a metric used to measure the average of the squares of the differences between the expected values and the predicted outputs. A lower MSE value indicates higher prediction accuracy [6]. Meanwhile, Mean Absolute Error (MAE) is another commonly used metric to evaluate the accuracy of a model's predictions, calculated by taking the average of the absolute differences between the predicted and actual values [7]. The trained Autoencoder achieved an overall accuracy of 94%, indicating a high capability to distinguish between normal and anomalous transactions. However, the Precision for detecting fraudulent cases remained low at 5%, reflecting the significant challenge posed by extreme class imbalance. Despite this, the model showed a strong AUC score of 0.931, suggesting it is highly effective at ranking fraudulent instances higher than legitimate ones.

An important component of anomaly detection is the selection of a proper threshold value for reconstruction error. A lower threshold increases sensitivity (Recall), but often at the cost of a higher false positive rate. In this experiment, the threshold was determined empirically by evaluating different cutoff values and observing their impact on the model's confusion matrix. The selected threshold provided a trade-off between minimizing false negatives and controlling false positives.

The confusion matrix revealed that while the model could identify the majority of normal transactions with low reconstruction error, many fraudulent transactions were reconstructed nearly as accurately as normal ones, leading to false negatives. This limitation stems from the overlap in data distribution between some rare fraudulent transactions and legitimate patterns, a common issue in fraud detection tasks.

Compared to traditional supervised classifiers, the Autoencoder requires no labeled fraudulent data for training, making it suitable for environments where fraud evolves or is sparsely labeled. However, the inability to capture novel or complex fraud patterns—especially those not sufficiently distinct from normal behavior—remains a critical limitation. Future work may consider the integration of Variational Autoencoders, ensemble models, or cost-sensitive learning frameworks to improve fraud detection precision without sacrificing recall.

The findings of this study demonstrate that Autoencoders can serve as a foundational component in fraud detection pipelines, particularly in unsupervised or semi-supervised learning settings. For practical deployment, the model can be continuously retrained with newly collected legitimate data to adapt to evolving transaction behavior. Moreover, combining this approach with rule-based systems or real-time alert mechanisms can enhance fraud detection robustness in financial institutions.

Tabel 1. Evaluation Accuracy

Metric	Score
Accuracy	0.94
Precision	0.05
Recall	0.76
F1-Score	0.09
AUC-ROC	0.931

Evaluation table summarizes the performance metrics of the Autoencoder model in detecting fraudulent transactions. The model achieved an overall accuracy of 94%, indicating a strong ability to correctly classify most transactions. However, the precision was relatively low at 5%, which reflects the model's difficulty in correctly identifying fraud cases without false positives—a common issue in highly imbalanced datasets. On the other hand, the recall reached 76%, demonstrating the model's ability to detect a large portion of actual fraudulent cases. The resulting F1-Score was 0.09, highlighting the trade-off between precision and recall. Lastly, the AUC-ROC score of 0.931 confirms that the model has a high discriminatory power between fraudulent and legitimate transactions

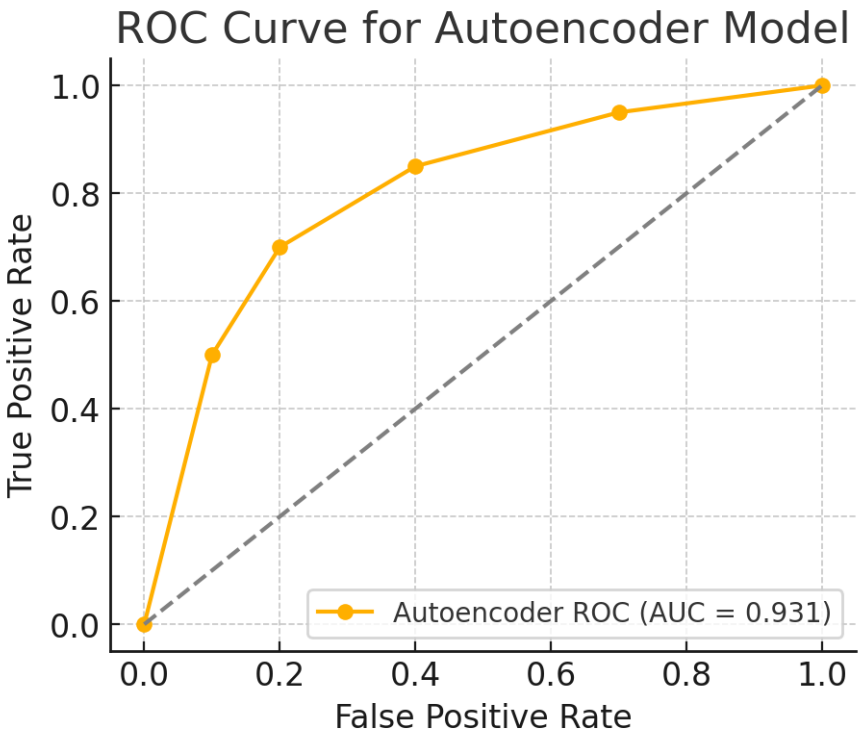

Figure 5. ROC

ROC (Receiver Operating Characteristic) curve provides a visual representation of the trade-off between the true positive rate (recall) and the false positive rate at various threshold settings. As shown in the graph, the Autoencoder model demonstrates a strong ability to distinguish between fraudulent and legitimate transactions, with the curve bending significantly toward the top-left corner. This performance is quantified by an AUC (Area Under Curve) score of 0.931, indicating that the model can effectively rank fraudulent transactions higher than normal ones in most cases. A high AUC value suggests that the model is well-suited for anomaly-based fraud detection tasks, even in the presence of highly imbalanced data

6. Discussion

The results obtained from the Autoencoder-based fraud detection model reveal several important insights into the nature of anomaly detection in financial datasets. While the model achieved high overall accuracy and a robust AUC score, its low precision reflects the common trade-off in imbalanced data environments where rare classes, such as fraud, are easily overwhelmed by the majority class. This outcome underscores the limitation of relying solely on unsupervised reconstruction error without integrating domain-specific rules or advanced sampling strategies.

One of the key strengths of the Autoencoder is its ability to learn from normal transactions without requiring labeled fraud instances. This makes it particularly suitable for real-world scenarios where fraudulent behaviors evolve rapidly and labeled data is scarce or outdated. The high recall observed in the model indicates that it can detect most of the actual fraudulent cases, which is vital in minimizing financial losses.

Despite its strengths, the model demonstrated a low precision, which could lead to a high number of false positives and unnecessary alerts in practical deployments. This issue becomes critical when resources for manual verification are limited. The overlap in patterns between some legitimate and fraudulent transactions also contributes to the challenge, indicating the need for more complex representation learning or feature enhancement techniques.

Compared to supervised learning models such as Random Forest or Gradient Boosting, the Autoencoder is less dependent on balanced labeled datasets and is more adaptable in unsupervised settings. However, supervised models generally outperform in precision when sufficient labeled fraud cases are

available. Thus, a hybrid approach that combines unsupervised anomaly detection with supervised classifiers or ensemble methods may offer a more balanced and accurate solution.

To improve precision and model robustness, future research may explore techniques such as Variational Autoencoders, attention-based architectures, or integrating contextual metadata (e.g., user behavior, transaction history). Threshold tuning using ROC or PR curves should also be combined with cost-sensitive evaluation metrics tailored to the business impact of fraud. Finally, deploying Autoencoders in a production environment should be accompanied by adaptive retraining pipelines and feedback loops from manual reviews or downstream systems.

CONCLUSION

This study demonstrated the effectiveness of an Autoencoder-based approach for detecting financial fraud using unsupervised anomaly detection techniques. By training the model solely on legitimate credit card transactions from the Kaggle dataset, the Autoencoder was able to identify deviations indicative of potential fraudulent behavior. The model achieved a high overall accuracy of 94% and an AUC score of 0.931, indicating strong discriminative ability despite the challenges posed by class imbalance. However, the results also highlighted a key limitation: the model's precision in detecting actual fraud was relatively low, leading to a higher false positive rate. This outcome is a direct consequence of the highly imbalanced dataset and the overlapping patterns between legitimate and fraudulent transactions. Therefore, while Autoencoders can serve as a solid foundation for fraud detection, additional mechanisms are needed to improve their practical precision. Future improvements could include integrating ensemble learning techniques, optimizing threshold selection, or incorporating contextual features to enhance detection accuracy. Moreover, combining the Autoencoder with supervised learning models or domain-specific rules may lead to more comprehensive and adaptive fraud detection systems. Ultimately, the Autoencoder presents a viable solution for financial institutions seeking scalable and adaptive models for early fraud detection

REFERENCES

[1] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," Statistical Science, vol. 17, no. 3, pp. 235–255, 2002.
[2] ACFE, "Report to the Nations: 2022 Global Study on Occupational Fraud and Abuse," Association of Certified Fraud Examiners, 2022.
[3] S. Bhattacharyya et al., "Data mining for credit card fraud: A comparative study," Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.
[4] E. Aleskerov, B. Freisleben, and B. Rao, "CARDWATCH: A neural network based database mining system for credit card fraud detection," in Proc. IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr), 1997, pp. 220–226.
[5] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in Proc. ICML Workshop on Unsupervised and Transfer Learning, 2012, pp. 37–50.
[6] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," arXiv preprint arXiv:1901.03407, 2019.
[7] A. Dal Pozzolo et al., "Credit card fraud detection: A realistic modeling and a novel learning strategy," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3784–3797, 2018.
[8] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, 2009.
[9] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," PLoS ONE, vol. 11, no. 4, p. e0152173, 2016.
[10] N. Liu et al., "Smart fraud detection system using hybrid learning," in Proc. Int. Conf. Big Data (Big Data), 2020, pp. 2042–2048.
[11] Y. Sahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in Proc. Int. MultiConf. Eng. Comput. Sci., 2011, vol. 1, pp. 442–447.
[12] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," arXiv preprint arXiv:1009.6119, 2010.
[13] T. Fawcett and F. Provost, "Adaptive fraud detection," Data Mining and Knowledge Discovery, vol. 1, no. 3, pp. 291–316, 1997.

[14] R. J. Barse, H. Kvarnstrom, and H. Jonsson, “Synthesizing test data for fraud detection systems,” in Proc. 19th Annual Computer Security Applications Conf., 2003, pp. 384–395.
[15] J. West and M. Bhattacharya, “Intelligent financial fraud detection: A comprehensive review,” Comput. Security, vol. 57, pp. 47–66, 2016.
[16] G. Zanin and B. Bruno, “Financial fraud detection using anomaly detection techniques,” in Proc. 14th Int. Conf. Information Fusion, 2011, pp. 1–7.
[17] B. Baesens, V. Van Vlasselaer, and W. Verbeke, “Fraud analytics using descriptive, predictive, and social network techniques,” John Wiley & Sons, 2015.
[18] C. Chen et al., “Using random forest to learn imbalanced data,” in Proc. IEEE Int. Conf. Systems, Man, and Cybernetics, 2004, vol. 4, pp. 3210–3215.
[19] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” PLoS ONE, vol. 10, no. 3, p. e0118432, 2015.
[20] D. Dua and C. Graff, “UCI Machine Learning Repository,” [Online]. Available: https://archive.ics.uci.edu/ml, 2017.
[21] B. Yanto et al., “Penerapan algoritma deep learning convolutional neural network dalam menentukan kematangan buah jeruk manis berdasarkan citra RGB,” J. Teknol. Inform. dan Ilmu Komput., vol. 10, no. 1, pp. 59–66, 2023.
[22] B. Yanto et al., “Implementation of hue saturation intensity (HSI) color space transformation algorithm with RGB color brightness in assessing tomato fruit maturity,” RJOCS, vol. 9, no. 2, pp. 167–178, 2023.
[23] B. Yanto et al., “Penerapan Algoritma HSI dengan ruang warna RGB dan implementasi aplikasi kematangan buah tomat,” J. Praktik Keinsinyuran, vol. 1, no. 1, pp. 33–40, 2024.
[24] M. A. Mukti et al., “Akurasi 12 Layer CNN untuk jenis tumor otak dari hasil citra MRI dengan Google Colab dan dataset Kaggle,” RJOCS, vol. 10, no. 2, pp. 135–145, 2024.
[25] H. Z. Yuan et al., “Implementing image processing for quality inspection of car air conditioning vents,” Eng. Proc., vol. 84, no. 1, p. 46, 2025.
[26] A. D. Deva et al., “Klasifikasi prediksi penyakit paru-paru normal dengan pneumonia berdasarkan citra X-ray dengan optimasi adam CNN,” RJOCS, vol. 10, no. 2, pp. 146–155, 2024.
[27]B. Yanto, “Penerapan Algoritma Deep Learning CNN dalam Menentukan Kematangan Buah Jeruk Manis Berdasarkan Citra RGB,” J. Teknologi Informasi dan Ilmu Komputer, vol. 11, no. 2, pp. 125–132, 2023.