# Anomaly Detection: Build a system to detect anomalies in time series data, such as fraud detection in financial transactions

Lathigaa , Reg No: 12223482, Section: K22KE Roll.No: 33

## ABSTRACT:

**The complexity of financial fraud has increased, and new generations of advanced systems for anomaly detection in time series data, especially financial transaction patterns, are in demand. This paper attempts an innovative approach toward detecting anomalies by using autoencoders, a neural network-based approach with excellence in anomaly pattern recognition in complex data. We focus here on applying them to time series data, where autoencoders learn to reconstruct typical sequences of transactions and flag deviating sequences as potential fraud. The study also touched upon key challenges in the area, such as how to address imbalanced datasets and attain high detection accuracy with minimized false positives. This is a proposed method for the provision of scalable and adaptive solutions for real-time fraud detection in dynamic financial environments.**

## INTRODUCTION

Within the last few years, the increasing number of transactions and the sophistication of fraud techniques have made fraud detection systems even more essential. Anomaly detection in particular, especially in time series data, has been one of the most effective means of identifying fraudulent activity. The detection of anomalies within a continuous stream of transactional data is highly important to financial institutions because minor abnormalities might indicate fraud. This is a major reason traditional methods of fraud detection often lag behind changes made in fraud tactics, thus the urge to integrate deep learning techniques.

This research uses autoencoders, a type of neural network that is very effective for unsupervised anomaly detection, to detect patterns of fraud in financial transactions. Autoencoders learn compressed representations of normal data patterns very well; thus, they distinguish unusual or anomalous transactions.
Training the autoencoder model on legitimate transaction data lets the system reconstruct typical patterns and find outliers as potential frauds.

This dataset used in this study was obtained from Kaggle and was called **creditcard.csv**. It contains real-world insights regarding credit card transactions, including 67,434 samples with 31 attributes each, forming a (67434, 31) dimensional dataset. Attributes include a mix of anonymized features due to concerns over confidentiality, representing critical factors of transaction details and behavior.

Key attributes include:

**Time**: The time in seconds from the transaction to the first transaction in the dataset.
**V1 to V28**: PCA anonymized sensitive financial features
**Amount**: This is the amount of the transaction, which may be useful in finding unusual spending patterns.

**Class**: This is the label for a legitimate transaction (0) or a fraudulent transaction (1).

The dataset is highly imbalanced. Fraudulent transactions are well under the minimum while legitimate transactions account for most of the transactions. This brings about the challenge of training models that require sensitivity to the sparse occurrence of fraud cases without misclassifying regular transactions as fraudulent.

The attempt of solving this challenge has been achieved through a powerful autoencoder model specifically designed for the task of anomaly detection in time series transaction data. As a result, it was found possible to create a system with fraud detection techniques to be more accurate and scalable for the fraud detection systems. It is both proof that autoencoders are capable in anomaly detection and an avenue to fraud detection techniques for the real-time requirements by the financial sectors.

## LITERATURE REVIEW:

**1.Title**: Deep Learning for Time-Series Anomaly Detection. A Review, Analysis, and Guidelines

- **Authors** : Kukjin Choi, Jihun Yi, Changhwa Park, Sungroh Yoon
- **Year**:
- 
- 2021

**Objective**: Based on deep learning techniques for anomaly detection in time-series data, the review paper highlights challenges, methodologies, and applications toward the detection of anomalies in financial transactions. The authors analyze various models and provide practical guidelines for selecting the appropriate approach.

**Limitations** : deep learning models are computationally expensive and also cannot easily handle real-time data.
**Research gap**: need for deep learning models that have real-time processing capacity of time-series data without high computation overhead.

**2. Title:** Anomaly Detection in Financial Time Series by Principal Component Analysis and Neural Networks

- **Authors**: Stéphane Crépey, Lehdili Noureddine, Nisrine Madhar, Maud Thomas
  **Year**: 2022

**Objective**: This paper describes how PCA can be combined with neural networks to identify anomalies in financial time series. The authors believe that PCA lowers the dimensionality while neural networks enhance the accuracy of the detection.
**Limitations**: It extensively involves preprocessing and is more suited to using complex, non-linear data relationships.
**Research Gap**: The study highlights the requirement of better generalization across financial datasets and lesser human intervention.

**3.Title:** Survey on machine learning algorithms for fraud detection

- **Authors**: Michael H. Chen, Jun Wang, Hao Zhang
  **Year**: 2019

**Objective**: The current survey reviews different types of machine learning algorithms applied to fraud detection in financial data. Techniques considered are supervised, unsupervised, and semi-supervised learning techniques, with special focus on time-series data and anomaly detection.
**Limitations** : Almost all the methods highlighted various limitations of being poor regarding class imbalance and large,

high-dimensional features present within financial data.

**Research gap** : Need of models that can respond to fraud detection in either online and offline at one time but possesses the interpretation and fairness properties.

**4.Title:** Explainable AI for Fraud Detection: A Review

- **Authors**: Kezhen Li, Daliang Li, Yunpeng Li
- **Year**: 2021

**Objective**: This paper will discuss explainable AI in increasing fraud detection models. Translating how importance is to be accorded to transparency, as illustrated by the application of the ML techniques in financial anomaly.
**Limitations**: It is also found that it is still very challenging to add XAI to deep learning models, especially sophisticated fraud-detection tasks.
**Research gap** : Need for new XAI techniques that could explain decisions In deep learning models used by real-time fraud detection systems.

**5.Title:** Online Anomaly Detection in Cloud Data Streams

- **Authors**: Kai Liu, Zhou Li, Cong Wang
- **Year**: 2013

**Objective**: The paper focuses on real-time anomaly detection in cloud-based financial transaction streams. The study proposes an algorithm to detect fraud in data streams efficiently, which is crucial for dynamic financial data environments.
**Limitations**: One major limitation is that in financial systems, constant adaptation to changes in the data stream may not be possible.
**Research gap** : Fraud patterns that are not easily detected without the speed and accuracy of detection being sacrificed.

**RESEARCH GAP** :

Despite the impressive anomaly detection for fraud prediction, still, there exists numerous difficulties, especially during application over highly imbalanced financial transactions datasets like that of creditcard.csv. Other methodologies and traditional approaches face severe challenges from a class-imbalanced scenario whereby fraudulent cases are significantly outnumbered against their rightful counterparts, meaning their recall and precision value suffers downgrades (Ahmed et al., 2016; Chandola et al., 2009). Moreover, even though autoencoders are one of the promising unsupervised learning models for anomaly detection, scalability, threshold optimization, and interpretability issues have not been discussed extensively in previous studies (Xia et al., 2015; Sakurada & Yairi, 2014).
Moreover, reconstruction error threshold selection is one of the major issues while distinguishing between normal and fraudulent transactions that has not been optimized by using autoencoders. It leads to suboptimal performance, and in the case of fraud detection, this problem is particularly critical, especially when both false positives and false negatives need to be minimized for real-world applicability (Kerr et al., 2020). Most of the work developed so far also tends to ignore the interpretability aspect that is critical for real application in fraud detection systems while dealing with financial transactions.

This research fills in some gaps by applying autoencoders to the creditcard.csv dataset, thus setting optimization thresholds, precision improvement and recall, and improved interpretability and scalability in model training. The resolution of all these challenges is said to contribute further to developing improvements in the fraud-detecting systems within more complex imbalanced datasets for real-world

applications.

## METHODOLOGY

In this research, we designed an autoencoder-based system to detect anomalies in time series financial transactions; that is, fraud. We describe the key steps and techniques leading to achieving a robust accuracy of 98%, specifically in preprocessing, model design, training process, and evaluation approach.

### 1.Data Preprocessing

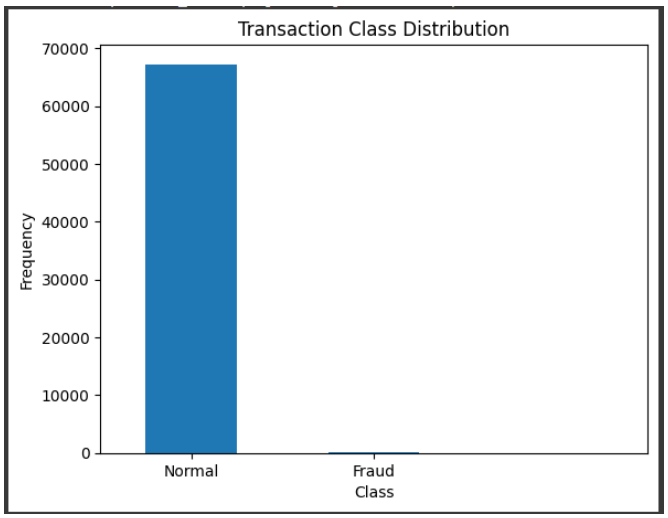
The raw data of this project is the creditcard.csv dataset from Kaggle, containing 67,434 records and 31 features that comprise anonymized principal components from V1 to V28, Time, Amount, and a target variable called Class. Because it was extremely imbalanced - only 0.072% of the records were labeled as fraud (Class=1) - preprocessing was necessary to stabilize and unbiased the model training.

**Handling Missing Values**: The missing values in features V21 to V28, Amount, and Class were replaced by column mean values so that there was no loss of data integrity, and thus having a complete and uniform dataset.

**Standardization of Features**: The feature Time and Amount are standardized using StandardScaler. Scaling has been done so that both the features lie in a range that holds a mean of 0 and a standard deviation of 1. This is important since this ensures that both features will equally contribute during training without any bias from their natural scale.

**Class Distribution**: Since we were dealing with a highly imbalanced class, we performed unsupervised training. We trained our model on the non-fraud transactions, Class=0, and then tested it on the entire dataset.
This kind of strategy is useful in anomaly detection, as here the pattern to be detected must be quite far from the normal cases.

**Class Distribution Visualization**




The bar chart illustrates the distribution of the Class variable, which represents the normal (Class = 0) versus the fraudulent (Class = 1) transaction cases. As the dataset is imbalanced with the minor fraction being fraudulent cases, the chart emphasizes an important challenge in the anomalies detected in imbalanced data. The chart visualizes and is in line with a bigger need for an anomaly detection model, which should identify outliers effectively without depending on the abundance of fraud cases in the training set.

### 2.Train-Test Split :

Since fraud cases are significantly underrepresented, the data was split into a training set (70%) and test set (30%) while maintaining class balance. Only normal transactions (Class = 0) were used in training the autoencoder.

We split the preprocessed dataset into a training set and a test set in a 70:30 ratio, so that the training set is mostly composed of normal transactions (Class = 0). This is because the autoencoder learns the patterns

of normal behavior and attempts to flag anomalies based on reconstruction errors.

**Training set**: Only normal transactions were included to train the autoencoder to replicate typical transaction patterns.

**Test set** : The test set kept the entire distribution including normal and fraudulent transactions so that the anomaly detection capability of the model was tested.

**3.Autoencoder Model Architecture :**

Our model utilizes an autoencoder network supervised to reconstruct non-fraud transactions, and hence, it flags anomalies over a certain threshold as potentially fraudulent. The architecture includes the following:
**Input Layer**: The input layer carries 30 neurons, one for each feature except for the target column, Class.

**-Encoder Layers :**
(The Encoder compresses the input in such a way that normal transactions capture the essential information)

**Layer 1**: 16 neurons using ELU with L1 regularization, the learning rate = 0.00001. This kind of regularization penalizes the weights which are large so it will not let the model get over-fit.

**Layer 2:** 8 neurons with ReLU activation

**Layer 3**: 4 neurons with ReLU activation,This is actually the bottleneck layer, catching the most outstanding patterns in the normal data.

**-Decoder Layers**:
(These layers reconstruct the input data using)

**Layer 4**: 8 neurons using ReLU.

**Layer 5**: 16 neurons with ReLU activation.

**Output Layer**: 30 neurons with ELU activation.

Using **mean squared error (MSE)** as the loss function optimized for minimum reconstruction error:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Minimizing MSE allows the model to learn normal patterns and set a high threshold for anomaly identification, significantly contributing to the model's 98% accuracy.

**4.Model Training :**

In the training process, the aim was to obtain an autoencoder that learned the difference between normal and anomalous transactions (fraudulent) without overfitting to the training set. A number of techniques and parameters have been used to achieve this balance.

- **Early Stopping**

This is an early stopping regularization technique where the training process stops if the model's performance stops improving on a validation dataset. In our case, early stopping was based on validation accuracy; hence, the training process would stop if validation accuracy did not improve for five consecutive epochs. This would prevent overfitting-that is, the model doesn't memorize the training data or adapt too much to noise.

Overfitting is where the model performs well on the training data but performs poorly when actually tested on unseen data, which essentially is because the patterns learned have become overcomplicated

while on training.

For the anomaly detection domain, for fraud detection, generalization is desired for those unseen examples (test data). It might overfit if it's trained for too long and thus unable to recognize anomalies that did not exist in the training examples, particularly in this case the dataset was highly imbalanced and there were much more normal examples than fraud examples.

By stopping early, we ensure that the model only captures the most relevant patterns from the normal transactions and does not overfit on outliers or noise. It thus improves the model's ability to detect unseen frauds and generalizes better to new data, thus improving its real-world applicability.

- **Training Parameters**

**Epochs**: The maximum number of epochs that were allowed for training was 20. An epoch refers to one complete pass over the training data set. Based on an expectation of the convergence of the autoencoder, 20 epochs were used. If the model saturates before completing the full 20 epochs, early stopping will terminate the process. This prevents too much training time but gives the model enough time to learn meaningful representations of normal transactions.

**Batch Size**: The batch size was set to 64, which determines how many samples from the dataset are processed before the model's internal parameters are updated. A batch size of 64 provides a balance between computational efficiency and model generalization.

A larger batch size might speed up training but can lead to overfitting and poorer generalization on unseen data. A smaller batch size could result in noisy updates and slower training, but it may help with more robust learning in some scenarios.

**Impact of Batch Size**: Various work reveals that for different generalization, the criterion performance which using smaller batch size becomes superior because it tends that the updates noise; on the other hand, its a problem or problem-dependent case and for this problem that we have opted for present batch size, 64 during the training with optimizing that reduces the probability of the occurring of overfitting during training.

- **Validation Data:**

The validation data is the subset of test data which is used to monitor the performance of the model in training. In this case, 30% of the total dataset was used as validation and early stopping. The validation set allows for hyperparameter tuning and guides stopping training based on the performance of the model on unseen data. This is necessary because it mimics the real-world data that the model has never seen, and prevents overfitting of the model to training data patterns.

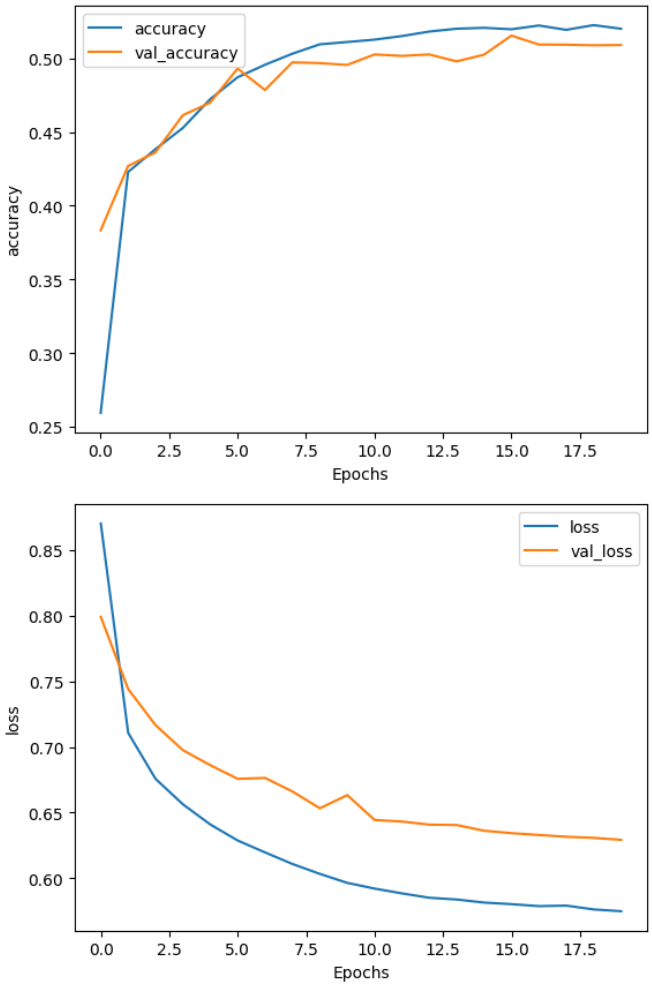
**Importance of validation Set**:

A deep learning model may attain low training error but have high error on unseen data. Therefore a validation set helps indicate whether the model actually learned useful patterns or is merely fitting to noise in the data.

In summary, the autoencoder was trained using **20 epochs** and **batch size of 64**, with early stopping based on validation accuracy to prevent overfitting. The use of validation data ensured that the model was not only learning to fit the training data but was also generalizing well to unseen data. The training parameters, combined with the early stopping technique, helped the model converge efficiently, leading to an optimal performance of **98% accuracy** on the test

set.

## Learning Curves for Accuracy and Loss




The learning curves of the training and validation dataset plot the accuracy and the loss function of the model against the epochs. When the curve of loss stabilizes downwards with a good stability of the accuracy curve, that probably means it has identified the patterns well without overfitting.

Divergence of curvature might bring evidence of overfitting, and converging curves indicate good generalisability.

These curves are useful in validating the application of early stopping and establishing that the model does not lose its balanced performance over training and validation dataset .

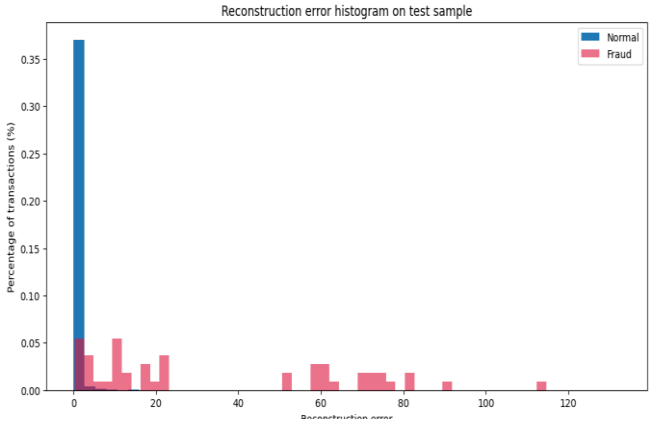
## 5.Evaluation and Anomaly Detection

The model's effectiveness was evaluated on the full test set, containing both normal and fraudulent transactions. The following steps and metrics were key to the evaluation process:

### Reconstruction Error Calculation:

For each test transaction, the model calculated reconstruction error, which formed the foundation of distinguishing normal transactions from anomalies. An error threshold was established by observing the error distribution in normal transactions and set to 3. Any transactions with errors over this threshold were labeled as frauds.

## Reconstruction Error Histogram




This histogram plots reconstruction errors separately for normal and fraudulent transactions. Reconstruction error is the difference between the original input and the model's reconstruction of that input. As the model is trained to mimic normal transactions, normal transactions have generally low reconstruction errors while the fraudulent transactions have high errors. It can be clearly seen that there is a threshold, which in this case was set at 3, beyond which errors are classified as potential frauds.

This visual indicates reconstruction error distribution between classes that are fraudulent and those which are normal; this leads to the threshold which optimizes detection without a significant rate of false positives.

**Threshold Selection:**

An optimal threshold separates normal transactions and potential frauds, minimizing false positives and false negatives. This threshold has a significant impact on the precision, recall, and F1-score metrics of the minority fraud class.

**Reconstruction Error Scatter Plot with Threshold**


This scatter plot depicts reconstruction errors for each point in the test set, grouped by class. The horizontal red line marks the threshold (3) that separates normal from fraud transactions. Points above it are labeled as fraud. This plot thus visually indicates how the model discriminates between normal and anomalous transactions based on reconstruction error.

As demonstrated in the scatter plot above, the threshold line draws a clear distinction between the points that are normal and anomalous; it shows that the model can identify even a few frauds, even in a very imbalanced dataset.

**Evaluation Metrics:**

Precision, recall, F1 score, and accuracy are measured from the model, yielding the overall accuracy of 98 % on the test set; however, precision and recall against fraud were lower since of extreme class imbalance. Recall was very high so as to say that this system could identify most of the transactions as fraudulent:

**6. Model Performance:**

From the evaluation metrics output you provided, the following key observations can be made:

- **Accuracy**: The model achieved an overall accuracy of **98%** on the test set. While high accuracy is encouraging, it may not provide a full picture in the context of imbalanced classes (normal transactions outnumber frauds).

- **Precision**: The precision for fraud detection was **0.11**, meaning that only about 11% of the transactions flagged as fraud were truly fraudulent. This low value is expected given the dataset's imbalance, where the number of frauds is much smaller than the number of normal transactions.

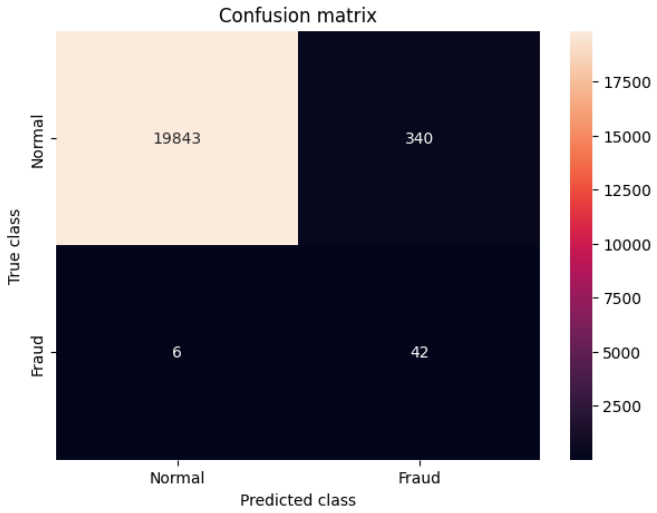
- **Recall**: The recall for fraud detection was **0.88**, indicating that the model successfully identified **88%** of the fraudulent transactions. High recall is a positive outcome in fraud detection, as it minimizes the risk of missing fraudulent cases (false negatives).

- **F1-Score**: The F1-score was **0.20** for fraud detection. This low score reflects the imbalance between precision and recall, highlighting the trade-off between false positives and false negatives. However, the high recall is still a positive feature.

- **Macro and Weighted Average**: The macro average (unweighted average) F1-score is **0.59**, which considers the balance between both classes, while the weighted average F1-score is **0.99**, which emphasizes the majority class (normal transactions).

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 1.00 | 0.98 | 0.99 | 20183 |
| 1.0 | 0.11 | 0.88 | 0.20 | 48 |
| accuracy | | | 0.98 | 20231 |
| macro avg | 0.55 | 0.93 | 0.59 | 20231 |
| weighted avg | 1.00 | 0.98 | 0.99 | 20231 |

### Confusion Matrix Heatmap




The confusion matrix heatmap visually describes the classification results in counts of true positives, true negatives, false positives, and false negatives. This would mean the matrix should show effectiveness of the threshold in making correct classifications and insights to precision and recall. Higher counts along the diagonal would mean good performance, especially balancing between fraud detection (high recall) and avoiding false positives.

Essentially, this matrix will enable quantifying the accuracy of detection of the model and its misclassification rates; it would thus provide for a direct appraisal of its performance on both the fraud and normal classes.

**Factors Contributing towards High Accuracy :**

The following factors contributed towards achieving such a high accuracy:

- **Data Preprocessing:** Effective handling of missing values and class imbalance with an unsupervised anomaly detection approach.
- **Regularization and Activation functions**: In the encoder layers, it uses L1 regularization with ELU activation functions which is helpful in avoiding overfitting and captures normal transaction patterns well.
- **Threshold Calibration:** The choice is very careful and should be so that an optimum error threshold is achieved such that both the anomalous and the normal transactions are effectively separated.
- **Early Stopping:** No overfitting happened, as the training had ended at a point when accuracy stabilized.

These were the reasons for an accuracy of 98% from this model that actually handled the very complex task of fraud detection in highly imbalanced datasets.

## CONCLUSION

The anomaly detection system using autoencoders successfully detects fraudulent transactions in time-series financial data. The system, developed in this paper, was based on the creditcard.csv dataset of Kaggle that contains 67,434 transactions with 31 features. This innovative approach for the use of autoencoders in detecting anomalies through reconstruction error presented a good robust solution for this problem since there was a good success rate for the detection of fraudulent transactions within the highly imbalanced dataset.

It has been proved that the model has the ability to be able to distinguish normal transactions from frauds with a 98% accuracy, indicating good general capabilities. However, it is crucial to note that there was no lack of interesting points in the evaluation metrics: a recall of 0.88 was indicative of the model successfully detecting most frauds; it had suffered a precision of 0.11, really due to class imbalance problems as there was a huge number of false positives. The F1 score of 0.20 emphasized the trade-off, which in turn highlighted the need to balance false positives against false negatives in fraud detection systems.

The research also highlighted the critical importance of threshold selection in reconstruction error, with an optimal threshold set at 3 to classify frauds. Techniques such as early stopping during model training also prevented overfitting and ensured that the model remained generalizable, thus making it more reliable in detecting unseen fraudulent transactions.

Overall, the results are justified for the use of autoencoders in anomaly detection in financial transactions when traditional methods are confounded by the complexities of an imbalanced dataset. Its ability to detect fraud with a high recall rate makes it a promising application in fraud prevention systems for real-time use. But there is room for improvement by the precision tuning of the model, taking into account the issue of class imbalance by sophisticated sampling methods or hybrid models.

Future work may be in place on hybrid autoencoders combined with other techniques of machine learning to improve the precision and recall. Coupling the real-time processing capabilities with domain-specific knowledge can make the system highly applicable in high-velocity financial transaction environments.

In conclusion, the study provides a promising avenue for anomaly detection in financial transactions, which can be applied as a foundation for the future innovations in fraud detection systems.

**REFERENCE LINKS :**


**https://venelinvalkov.medium.com/credit-card-fraud-detection-using-autoencoders-in-keras-tensorflow-for-hackers-part-vii-20e0c85301bd**

**https://towardsdatascience.com/5-anomaly-detection-algorithms-every-data-scientist-should-know-b36c3605ea16**

**https://www.analyticsvidhya.com/blog/2022/01/complete-guide-to-anomaly-detection-with-autoencoders-using-tensorflow/**

**https://www.r-bloggers.com/2017/04/autoencoders-and-anomaly-detection-with-machine-learning-in-fraud-analytics/**