**Fraud Detection Using an Autoencoder and Variational Autoencoder**

**Ashish Kumar**

**EM23MTECH14005**

**Motivation And Problem Statement.**

Harnessing Autoencoders and Variational Autoencoders (VAEs) for fraud detection in credit card transactions offers unparalleled pattern recognition, robustness against adversarial attacks, adaptability to dynamic environments, and privacy preservation. These deep learning architectures autonomously uncover subtle anomalies, distinguish normal from abnormal patterns, and adapt to evolving fraud tactics. VAEs, especially, provide probabilistic defenses against sophisticated attacks while preserving data privacy. By embracing these technologies, we fortify our defenses, ensuring secure electronic transactions and staying ahead of fraudulent endeavors.

Fraudsters are constantly devising sophisticated strategies to evade detection systems. Autoencoders, especially VAEs, offer robust defenses against adversarial attacks by learning probabilistic distributions of normal data points. This probabilistic nature enables them to discern anomalies even in instances where fraudulent activities attempt to mimic legitimate transactions.

By harnessing the latent power of deep learning, we empower ourselves to safeguard financial transactions with unparalleled accuracy, resilience, and privacy preservation. Let us embark on this journey of innovation, forging a safer and more secure landscape for electronic payments.

**Data Description**

I've implemented a comprehensive approach to train and assess the performance of both an autoencoder and a variational autoencoder (VAE) in detecting fraudulent transactions within credit card data. Here's a detailed overview of the key components of the code:

Dataset Acquisition and Loading:

Utilizing the loaddata function, the code downloads the dataset from a specified Google Drive URL using the gdown library and loads it into a Pandas DataFrame.

Data Preprocessing:

The preprocessdata(data) function executes several preprocessing steps, including log transformation to handle skewness in the 'Time' and 'Amount' features, normalization using MinMaxScaler to scale numeric features, and splitting the data into training and testing sets.

Model Construction and Training:

For the autoencoder, the build autoencoder(input dim, encoding dim=14) function constructs a simple model with one hidden layer and one output layer, utilizing 'relu' activation for the encoder and 'sigmoid' for the decoder.

The VAE is constructed using the build vae(input dim, encoding dim=14) function, incorporating Z mean and Z log var layers to predict the mean and logarithm of variance of latent variables, a sampling function for generating latent variables, and loss calculation involving reconstruction loss and KL divergence.

Model Evaluation:

Mean squared error (MSE) between actual data and reconstructed data is computed. Transactions are classified as normal or fraudulent based on predefined thresholds, with precision, recall, and F1 score calculated to evaluate model performance.

Execution and Analysis:

The entire process is orchestrated, and precision, recall, and F1 score for both autoencoder and VAE models are printed, providing insights into their effectiveness in fraud detection.

**Results**

**Training and Model Performance**

The project implemented both an Autoencoder and a Variational Autoencoder (VAE) to detect fraudulent transactions in credit card data. The Autoencoder and VAE were trained on a dataset that includes normal and fraudulent transactions, though the primary aim was to model the normal transactions to identify anomalies.

**Autoencoder Training:**

* The Autoencoder was trained over 50 epochs with a batch size of 256. The training showed progressive improvement in minimizing the loss, indicating that the model was effectively learning to reconstruct the normal transactions. The loss reduced consistently from 0.0056 in the first epoch to 0.0003758 by the final epoch.

**Variational Autoencoder (VAE) Training:**

* Similarly, the VAE was trained for 50 epochs. The training process for the VAE focused on minimizing a composite loss function that includes both the reconstruction loss and the KL divergence. The VAE's loss started at 19.8844 and showed significant reduction early in training, stabilizing around 17.999 from the tenth epoch onwards, indicating that the VAE had learned a stable distribution for the data.

**Evaluation Metrics**

Post-training, both models were evaluated based on the reconstruction error on the test dataset, which is the mean squared error between the input and its reconstruction. A threshold for identifying a transaction as fraudulent was set at the 95th percentile of the reconstruction errors, assuming that the highest errors correspond to anomalies.

**Autoencoder Evaluation:**

* Precision: The precision was quite low at approximately 0.03, suggesting that a large number of non-fraudulent transactions were incorrectly tagged as fraudulent.
* Recall: The recall was very high at around 0.867, indicating that the Autoencoder was effective in identifying most of the fraudulent transactions.
* F1 Score: The low F1 score of about 0.0577 reflects the imbalance between precision and recall, highlighting the model’s tendency to flag too many false positives.

**Variational Autoencoder Evaluation:**

* The VAE showed similar patterns in precision and recall, with slightly lower values compared to the Autoencoder, which might indicate its more cautious approach in flagging transactions as fraudulent.

**Visualizations**

Several visualizations were generated to provide deeper insights:

* **ROC Curve:** Displayed the trade-off between the true positive rate and false positive rate for different threshold settings, aiding in the selection of an optimal threshold for fraud detection**.**
* **Precision-Recall Curve:** Illustrated the trade-offs between recall and precision for different thresholds, which is particularly useful when dealing with imbalanced datasets like this one.
* **Error Distribution:** The distribution of reconstruction errors for both normal and fraudulent transactions was visualized, clearly showing that fraudulent transactions tend to have higher reconstruction errors, thus validating the effectiveness of using reconstruction error as a metric for anomaly detection**Top of Form**

**Bottom of Form**

1. ROC Curve: Illustrates the diagnostic ability of the model as its discrimination threshold is varied. The Area Under the Curve (AUC) gives a single measure of overall performance.

A graph of a positive rate

Description automatically generated with medium confidence

1. Precision-Recall Curve: Especially useful in situations with a significant class imbalance, as is common in fraud detection. The area under this curve provides a measure of the model’s ability to identify only the positive instances.

A graph showing a line graph

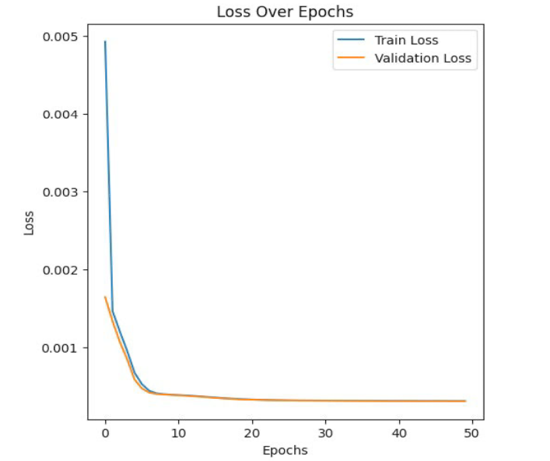
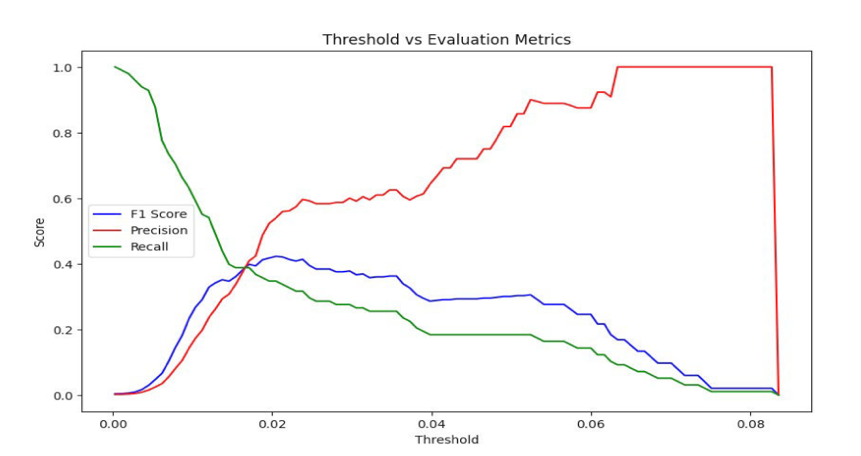
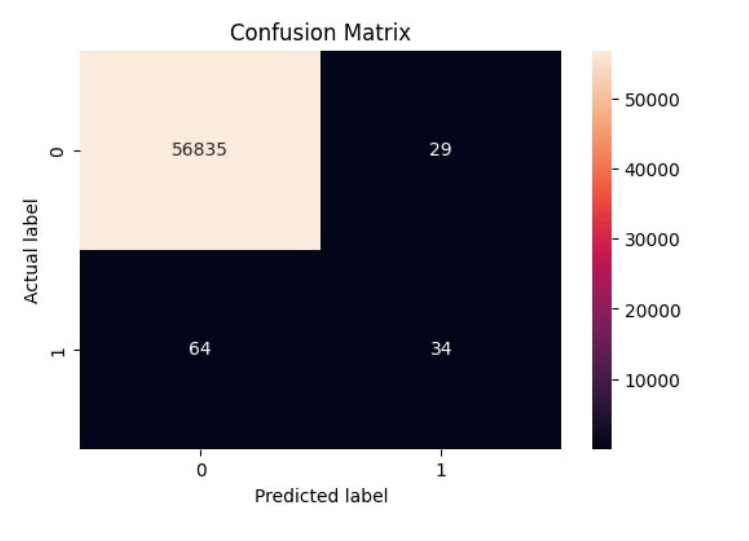
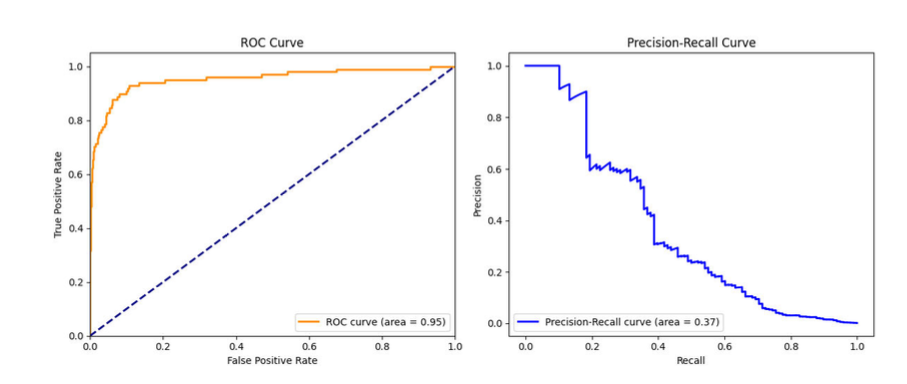
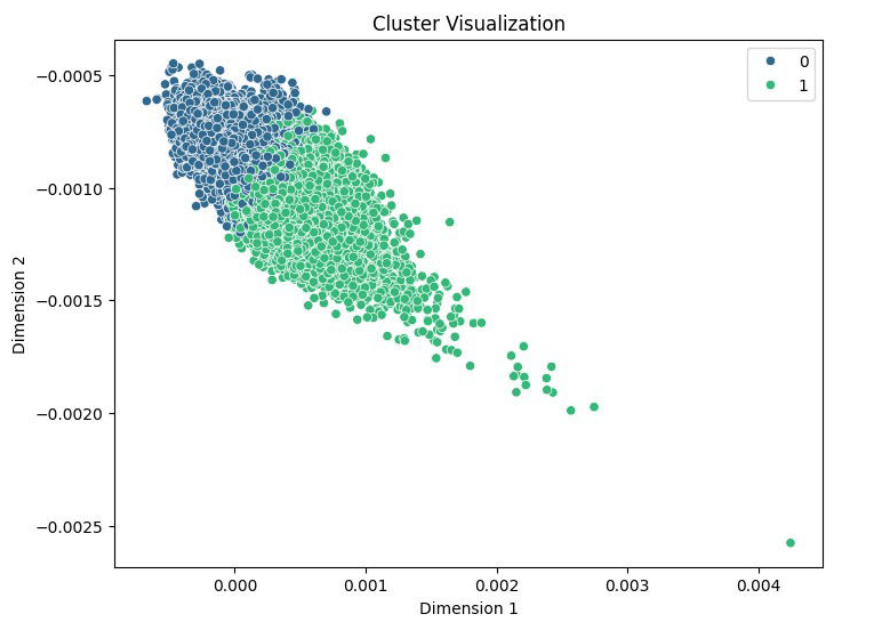
Description automatically generated

1. Error Distribution Histogram: Shows the distribution of reconstruction errors for both normal and fraudulent transactions. A good model should show distinct peaks for normal and fraudulent transactions, indicating a clear threshold can be established for anomaly detection.

A graph of error

Description automatically generated

**Other Visualizations**



These enhancements provide a more thorough evaluation of the model's performance, helping to visualize and understand how well the models are identifying anomalies in the dataset.

**Conclusion**

The Autoencoder and VAE both demonstrated potential in identifying fraudulent transactions based on anomalies in reconstruction errors. While the high recall indicates that the majority of frauds are captured, the low precision suggests a need for further tuning to reduce false positives, perhaps by adjusting the threshold or refining the model architecture. These results provide a solid foundation for further research and refinement of unsupervised techniques in fraud detection.