GEORGETOWN UNIVERSITY
The Graduate School of Arts & Sciences
Master of Science in Data Science & Analytics

Insurance Fraud Detection using Unsupervised Learning Techniques

Authors: Josh Sweren, Parsa Keyvani, Dizhou Zhang, Chongyuan Hong

DSAN 6600 Final Project Writeup

Introduction

Insurance fraud poses significant challenges for both insurers and consumers, leading to financial losses and inefficiencies in claims processing. According to a 2023 estimate from The Coalition Against Insurance Fraud, insurance fraud costs businesses and consumers $308.6 billion a year. This is a substantial increase from their $80 billion estimate from 1990, indicating that this issue continues to exacerbate[1].

As a result of this phenomenon, insurance companies are constantly searching for novel detection techniques. These often include non-analytical methods such as examining social behavior, personal identification irregularities, or insufficient documentation. Meanwhile, analytical fraud detection methods often rely on supervised learning, which requires labeled data that may be expensive and time-consuming to obtain.

This paper explores and identifies a use case in which one may not have reliable labeled data, and attempts to answer the question of whether deep learning can be used to singularly identify instances of fraud. Using a variety of unsupervised learning methods, we identify deviations in claims data to isolate anomalies and surmise that these might indicate fraud. We then assess our models' performance when applied to previously unseen data to further evaluate the real-world usefulness of anomaly detection as a means for detecting fraud.

Data

Two distinct datasets were used for model training and testing, originating from different sources and geographies. These datasets were selected to explore the feasibility of applying deep learning models in fraud detection.

The training dataset was sourced from the openICPSR repository, containing an administrative record of policy transactions from a Spanish motor vehicle insurance portfolio. This dataset covers the period from November 2015 to December 2018, with 105,555 rows and 30 columns. Each row represents a policy transaction, while the columns represent various attributes related to policy, vehicle, and demographic information. However, the training dataset lacks fraud labels, making it suitable only for unsupervised learning approaches. After preprocessing and data mapping, the training data included over 97,000 observations and seven features, such as Age, Claim_amount, and Vehicle_risk_type [6].

The testing dataset was obtained from Mendeley Data, consisting of policy data from a U.S.-based motor vehicle insurance company. This dataset includes over 1,000 rows and features

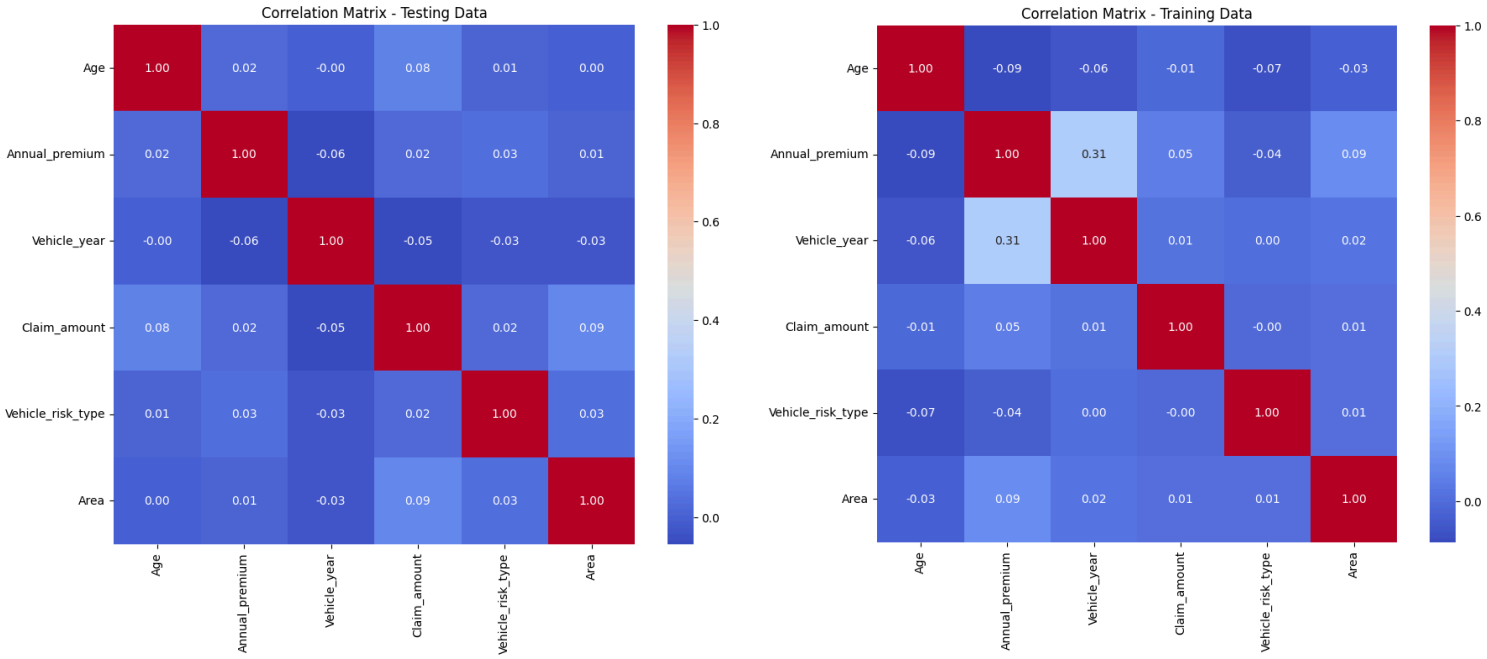
similar to the training dataset but also includes a fraud_reported label, which is critical for evaluating the model's accuracy in a supervised manner. After preprocessing and mapping to align the features with the training data, the testing dataset contained eight features, with the additional column being the fraud label [7].

Additionally, data mapping and feature selection were applied for consistency between the datasets. This process involved standardizing and transforming attributes such as *Policy_start_date* and encoding categorical variables like *Area* and *Vehicle_risk_type*. By creating a common data model, the final feature set for both datasets included shared attributes, enabling the application of unsupervised models on unified data. This approach allowed for training using the Spanish dataset and subsequent testing and evaluation using the U.S. dataset.

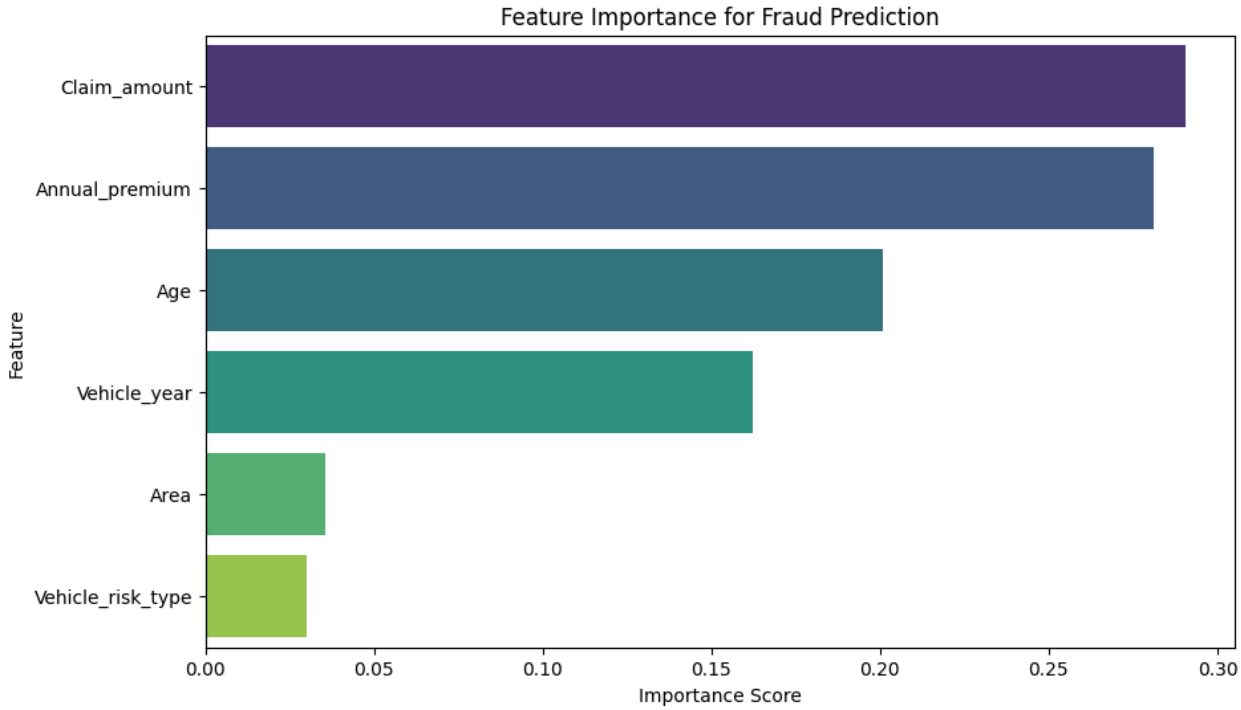
Feature Name	Description	Type	Dataset
Policy_start_date	The date when the insurance policy began.	Date	Train.csv & Test.csv
Age	Age of the policyholder at the time of policy start.	Numerical	Train.csv & Test.csv
Annual_premium	The annual premium amount paid for the insurance policy.	Numerical	Train.csv & Test.csv
Vehicle_year	The year the insured vehicle was manufactured.	Numerical	Train.csv & Test.csv
Claim_amount	The total amount claimed during the policy period.	Numerical	Train.csv & Test.csv
Vehicle_risk_type	The type of risk associated with the vehicle insured. Categories: 1 = Vans, 2 = Passenger Cars, 3 = Agricultural/Heavy Vehicles.	Categorical	Train.csv & Test.csv
Area	Indicates whether the incident occurred in a rural or urban area. Categories: 0 = Rural, 1 = Urban.	Categorical	Train.csv & Test.csv
fraud_reported	Indicates whether a fraudulent claim was reported. Categories: N = No fraud, Y = Fraud.	Categorical	Test.csv

Exploratory Data Analysis (EDA)

Our analysis examined the training and testing datasets to identify trends and relationships between the features and on the fraud detection task.



The correlation heatmaps above for both the training and testing datasets show weak relationships between most variables, with no exceptionally high correlations. The highest correlation observed was *Annual_premium* and *Vehicle_year* which showed a moderate positive correlation of 0.31 in the training data. This implies that more recent vehicles might be associated with higher premiums. The main takeaway from the above correlation heatmaps is that the variables contribute independently to the prediction task.



The feature importance chart above shows that Claim_amount is the most critical predictor of fraud, followed by Annual_premium, Age, and Vehicle_year. These features collectively show the potential for distinguishing between fraudulent and non-fraudulent claims. However, Area and Vehicle_risk_type exhibit lower importance scores, implying limited impact on prediction.

Fraud Reported	**Count**
No (N)	762
Yes (Y)	253

The above table shows the distribution of fraud labels in the test dataset. With 253 instances of fraud and 762 instances of no fraud, we can clearly see a class imbalance.

Methods

HDBSCAN

This project would focus on 4 specific ways on finding outliers as potential fraud, including HDBSCAN, Auto-encoder, GAN, and transformer. With different methods, we could cross-validate the effect of each method on outlier findings, and each unique approach could be suitable in different circumstances.

HDBSCAN is an advanced type of hierarchical clustering method. It can cluster datasets even under noise. It serves as the baseline of this fraud detection work compared to deep learning methods.

Auto-Encoder

Auto-encoder is a type of neural network that learns a compressed representation of data and reconstructs input, which makes it a perfect outlier detector. Generally, auto-encoders contain a singular fully connected hidden layer with both an encoder and decoder. During the encoding and decoding process, models learn patterns within the dataset and attempt to reconstruct the input data. The discrepancy between reconstructions and the actual input is defined as the “reconstruction error”, and records with a reconstruction error greater than a user-defined threshold are labeled as anomalies.

After hyperparameter tuning, the auto-encoder utilized activation functions of ReLU and Sigmoid for the encoder and decoder, respectively. The bottleneck size was 7, batch size was 64, and 50 epochs were specified, with early stopping. To update weights during training, the Adam optimizer was used, and the mean squared error was the loss function.

Generalized Adversarial Network

The goal of a Generative Adversarial Network is to find the potential outliers, taking into account the GAN capability of generating realistic data distributions. By training a Generator to produce samples similar to normal data and a Discriminator to differentiate between real and generated data, the model learns to detect anomalies as deviations from the expected data distribution. The confidence score for the output of the different samples by the Discriminator assigns the data as normal or abnormal by a certain threshold.

The GAN has a Generator and a Discriminator. The Generator takes in a latent noise vector and generates data samples similar to the normal distribution, utilizing LeakyReLU activation in its hidden layers and Tanh activation in the output layer to ensure that the range is [-1, 1]. The Discriminator, trained with LeakyReLU in its hidden layers and Sigmoid in the output layer, outputs a confidence score in the range [0, 1]. Both models are optimized using

Binary Cross-Entropy Loss. Discriminator is trained to distinguish between normal(from train.csv) and fraud data(from Generator), while Generator is trained to generate fraud data that can fool the Discriminator. A threshold set at the 30th percentile of confidence scores classifies samples as anomalies if their scores fall below the threshold. The model is trained on 50 epochs and with the Adam optimizer.

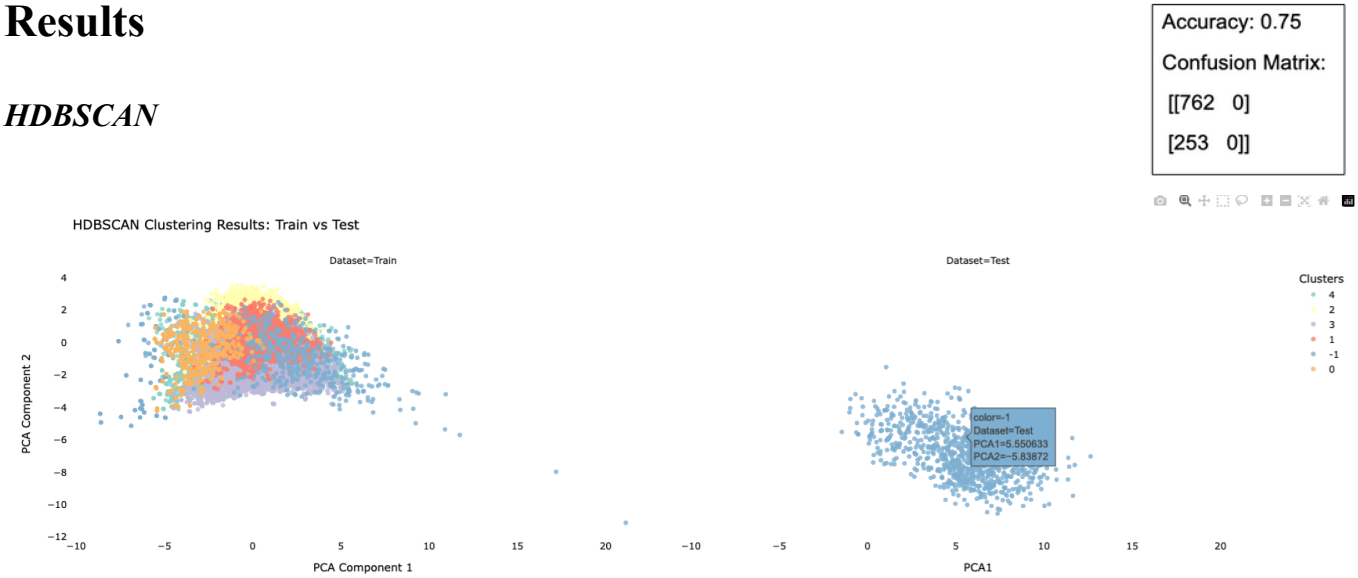
Transformer

Transformer is a type of neural network that contains a self-attention mechanism, allowing the encoder to learn data patterns and the decoder to predict based on it. This mechanism allows the model to capture dependencies between features, regardless of their order or position in the input. By using the threshold of sum probability output of each data, the model is used to find outliers of data in the dataset.

In this project, The transformer is being used to find outliers. This transformer is constructed by simple transformer architecture without positional embedding, which contains an embedding layer that converts 98(dictionary size) features of encoded input into output in 512 dimensions. Then we have a transformer layer which contains 8 heads with 3 encoded layers generating self-attention, following 3 decoded layers which predict based on encoded layer output, and return a 512 dimension output. At last, we have a linear layer that transformer 512 dimensions to 98, forming a logit output of dictionary size. Softmax is not needed since the torch classification method contains softmax.

Results

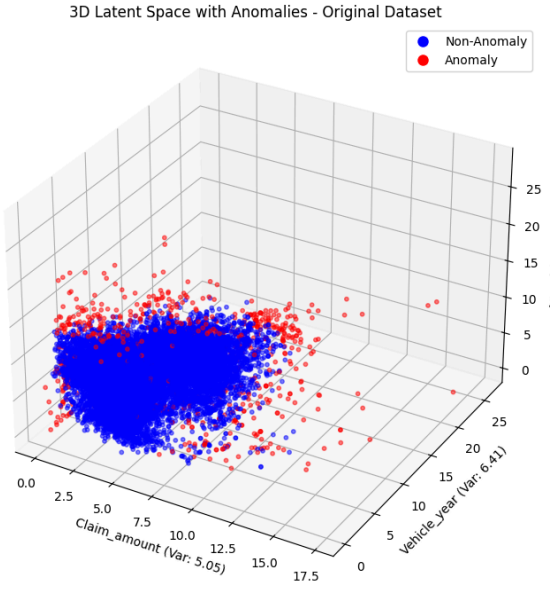
HDBSCAN



The HDBSCAN model yielded an accuracy of 75.1% on the test dataset. However, as visualized in the PCA-reduced plot above, all test points were assigned to Cluster -1, indicating they were classified as outliers. This outcome suggests that the clustering model did not generalize well to the test data, likely due to differences in feature distributions between the training and test datasets. The confusion matrix revealed that while all non-fraud cases (762) were correctly identified, the model failed to identify any fraud cases (253).

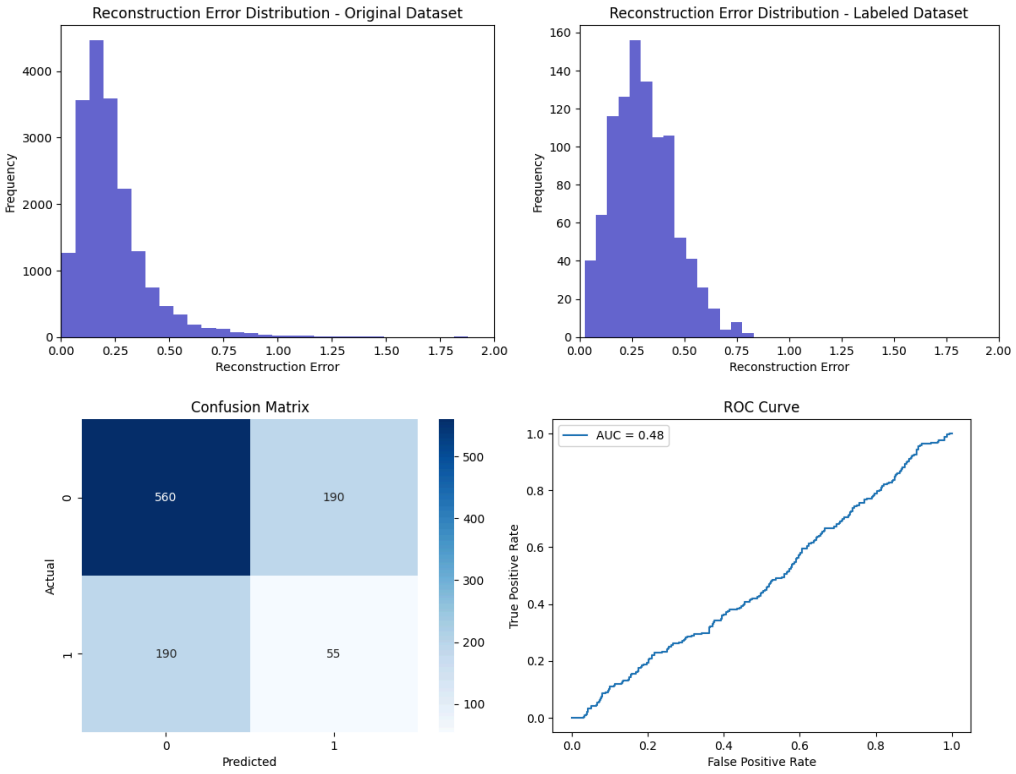
These results highlight the challenges of applying a density-based clustering model in a domain where the training and testing datasets differ significantly. Despite these limitations, HDBSCAN served as a useful baseline for comparison against more sophisticated deep learning models, which will be explored below to enhance fraud detection performance.

Auto-Encoder



The above image is a visualization of auto-encoder results using test data from the original dataset, with the axes being the latent spaces of the top three features in their contribution to variance. The purpose of showing this is to highlight the model's ability to identify patterns in the data, as anomalies (marked red) tend to stray from the rest of the data.

When the labeled data is introduced, this is when we find that the auto-encoder's ability to detect anomalies does not correlate with fraud detection. As we can see below in the first two visualizations, reconstruction errors, while still small, are noticeably larger than they were when testing on the original dataset.

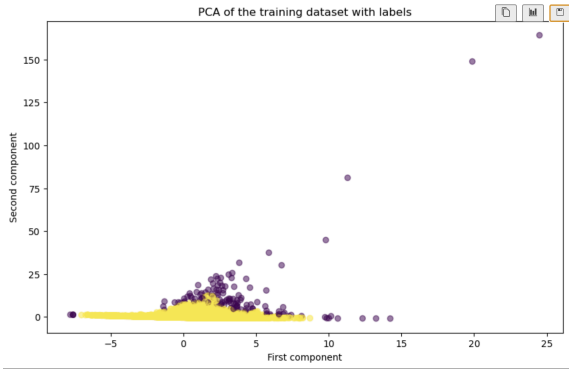


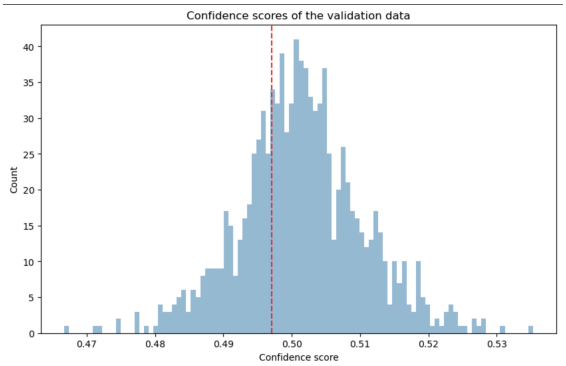
Meanwhile, the bottom two visualizations show performance when attempting to predict fraud by determining how many records categorized as anomalies lined up with those labeled fraud. The confusion matrix and ROC curve both suggest that the auto-encoder did this task no better than a random guess.

Ultimately, the auto-encoder was able to learn meaningful patterns within the data, as it managed to significantly diminish reconstruction error throughout training. However, these results show that this process did not produce accurate predictions of fraud, indicating that either the anomaly detection method has no alignment with fraud detection, or that the two datasets are too dissimilar to apply the training on one dataset to the testing on another.

Generalized Adversarial Network

The GAN model performed poorly, and it is better at classifying normal samples than anomalies. Maybe this is because of some imbalance in the training data. While the model correctly identified 71.05% of the normal samples, its anomaly recall was 33.2%, indicating that it missed many anomalies, which is quite worrying in various applications, such as fraud detection or even fault diagnosis. In the confusion matrix, 165 actual anomalies were classified as normal, while 218 normal samples were misclassified as anomalies. Accuracy also remains pretty low at 61.7% to show high difficulties in effectively separating the normal and anomalous data samples. These results show that the model has a bias in classifying data as normal.



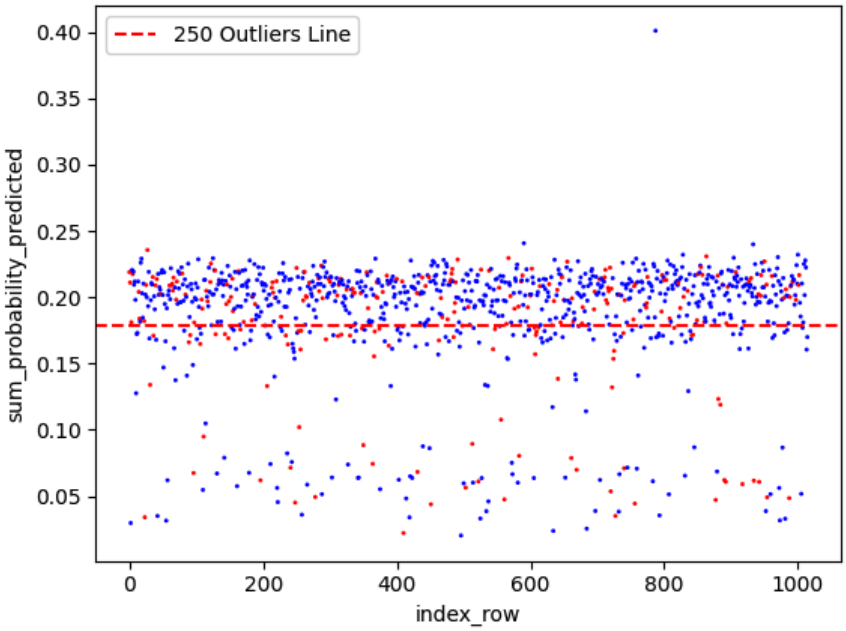


Confusion matrix	Predict Anomaly	Predict Normal
Actual Anomaly	82	165
Actual Normal	218	535

This project aims to demonstrate the potential of using Generative Adversarial Networks for anomaly detection, a confidence-based approach using the Discriminator for the identification of anomalous data. While the model worked reasonably well in classifying normal samples, the low anomaly recall shows the challenges that come with detecting rare, which is very crucial in fraud detection applications.

Transformer

The transformer uses 2-3 hours to finish 8 epochs training on a random sequence of labeled and encoded features of data each epoch, resulting in a model which can generate a probability distribution of prediction based on combination inputs, by sequentially pushing features columns, we can get the sum probability of each row of data through indexing with real data. We then use the 250 lowest probability sum line to judge whether a data row is an outlier. We use test data to check whether the outlier we predict is the same as real fraud data. The result is an accuracy of 0.657. The below image shows the test result, the red point is real fraud data.



Confusion matrix	True positive	True negative
Predict positive	76	171
Predict negative	177	591

Conclusion

Since our models did not perform well at detecting fraud, our discussion centered on anomaly detection and how it may not be a sufficient predictor of fraud, and whether unsupervised learning was suitable. We discussed some potential improvements, for example that we could choose a singular dataset with labels, so that we could do both supervised and unsupervised learning, while training and testing with the same data. By comparison, we can figure out which algorithm works best. Data with a clear fraud indicator will probably perform better data validation than training on data without one. Also, using data from two different companies could be painstaking and produce potential bias, which is not recommended to do according to our experience.

Despite having low accuracy in test data, each model does catch outliers and some are performing great. Auto-encoders, for example, catch many reasonable outliers in training sets. Models like GAN and transformer also trace some patterns instead of nothing, just not fitting with test data. Ultimately, the challenges found when attempting to train on unlabeled data, as

well as testing with a foreign dataset further emphasize the need to apply other techniques–analytical and non-analytical–to combat fraud.

References

1. Insurance fraud costs the U.S. $308 billion annually. ConroySimberg. (2023, March 17). https://www.conroysimberg.com/blog/insurance-fraud-costs-the-u-s-308-billion-annually/
2. Degirmenci, A., & Karal, O. (2022). Efficient density and cluster based incremental outlier detection in data streams. *Information Sciences*, *607*, 901-920.
3. Zhu, M., Gong, Y., Xiang, Y., Yu, H., & Huo, S. (2024, June). Utilizing GANs for fraud detection: model training with synthetic transaction data. In *International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024)* (Vol. 13180, pp. 887-894). SPIE.
4. Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, *479*, 448-455.
5. Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
6. Lledó, Josep, and Pavía, Jose M. *Dataset of an Actual Motor Vehicle Insurance Portfolio*. Inter-university Consortium for Political and Social Research [distributor], 8 Aug. 2023, https://doi.org/10.3886/E193182V1.
7. Mendeley Data. *A Dataset for Fraud Detection in Motor Insurance Claims*. Version 2, https://data.mendeley.com/datasets/992mh7dk9y/2, Accessed 10 Dec. 2024.