Applying Fine-Tuning methods to FTTransformer in Anti Money Laundering applications

Vasco de Graaff (V.P.deGraaff@student.tudelft.nl) Supervisors: Kubilay Atasu (kubilay.atasu@tudelft.nl) and Atahan Akyildiz(t.a.a.akyildiz@tudelft.nl)

1 Background

Detecting and preventing money laundering is essential for maintaining the integrity of global financial systems [2]. As criminal organizations develop increasingly sophisticated methods, enhanced systems are needed to effectively counteract these threats. This research explores fine-tuning performance on Feature Tokenizer Transformers (FTTs) [5] compared with Graph Neural Networks (GNNs) [4] for anti-money laundering (AML) applications.

Our research utilizes the IBM AML dataset [1], which provides a synthetic financial transaction dataset for benchmarking AML detection models. We aim to contribute to the development of more efficient and accurate methods for detecting financial fraud patterns.

Building on our baseline models provided by our supervisor, we investigate the effectiveness of fine-tuning techniques - LoRA [7], Vanilla fine-tuning and Freezing the model backbone. Previous research has demonstrated that fine-tuning pre-trained models can produce significant improvements for downstream tasks. By applying these techniques, we aim to develop more efficient and accurate methods for detecting financial fraud patterns, ultimately contributing to the advancement of AML technologies.

2 Research Question

- What is the effectiveness of supervised training? GNN vs FTT
- Explore Zero-shot (pre-training) performance
- Compare performance of pre-training + full fine-tuning vs pre-training + freezing model backbone
- How does LoRA fine-tuning perform?

3 Methodology

3.1 Supervised Learning

The first experiment investigates the performance of the baseline supervised learning model compared with the supervised GNN (PNA) model [4] reproduced. This comparison provides valuable insights into the suitability of the FTT model for handling the IBM AML dataset.

3.2 Self Supervised Learning

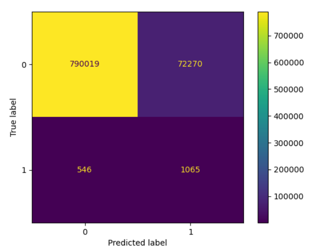
Next, we assess the effectiveness of pre-training(self supervised) strategies, specifically focusing on the impact of these strategies on a secondary task, which involves predicting the RMSE for the MCM task.

3.3 Fine-tuning

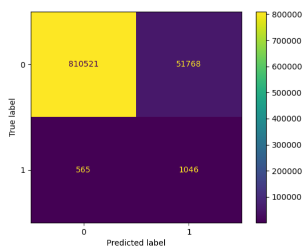
Finally, we explore the results of fine-tuning the self-supervised models on handling the classification tasks using Vanilla fine-tuning, Freezing the models backbone and applying LoRA fine-tuning.

4 Results

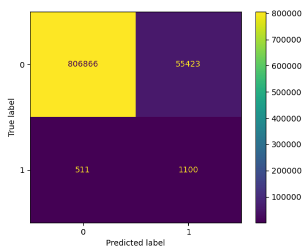
Our experiments yielded results that, while not meeting initial expectations, provided valuable insights into the challenges of applying tabular transformers and fine-tuning techniques to the IBM AML dataset.



Vanilla Fine-tune



Freeze Fine-tune



LoRA Fine-tune

Exploring our results, fine-tuning the self-supervised model did not outperform the original supervised model. The fine-tuned models gave low f1 scores (0.02840-0.0384), low precision score (0.0145-0.01981) and moderately high recall score (0.6493-0.6828). Compared to the 0.5603 which GNN(PNA) [4] reproduced from the AML paper achieved, shows that FTT lacks behind GNNs in this specific application. This can also be attributed to the pre-training process explored in the discussion in combination with the datasets properties.

5 Discussion

In the discussion of this poster, we explore the main challenges during these experiments. Further exploration can be found within the full report.

5.1 Dataset

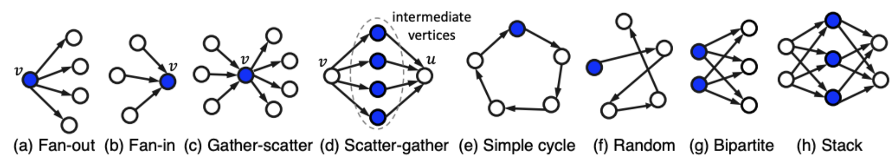


Figure: Synthesized Money Laundering Graph Structure from [1]

One of the key limitations of this project was the dataset's extreme class imbalance of 1000:1 (False to Positive examples), paired with the lack of discriminative attributes in the columns, hindered accurate predictions. Reconstructing the graph in the pre-training step, would've also have improved the fine-tuned models results.

5.2 LoRA

Though LoRA has proven to be extremely efficient at fine-tuning large language models[7], LoRA applied to our self supervised model, under-performed in the area of memory usage and training speed due to distribution of learnable parameters. This is because only around 5% of learnable parameters are in the backbone.

6 Conclusion

Our FTT for anti-money laundering using the IBM AML dataset, focusing on fine-tuning techniques. Key findings emphasized the importance of understanding dataset characteristics and fine-tuning had little performance improvements. Future work should address model limitations, explore graph reconstruction, and investigate techniques like oversampling [3], GNN fine-tuning, and comparisons with tree-based methods [6] to advance machine learning in AML applications.

References

[1] E. Altman, J. Blanuša, L. von Niederhäusern, B. Egressy, A. Anghel, and K. Atasu. *Realistic Synthetic Financial Transactions for Anti-Money Laundering Models*. 2024. arXiv: 2306.16424.

[2] N. Bakhshinejad, R. Soltani, U. Nguyen, and P. Messina. *A Survey of Machine Learning Based Anti-Money Laundering Solutions*. Oct. 2022.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (June 2002), pp. 321–357. ISSN: 1076-9757. DOI: 10.1613/jair.953.

[4] B. Egressy, L. von Niederhäusern, J. Blanusa, E. Altman, R. Wattenhofer, and K. Atasu. *Provably Powerful Graph Neural Networks for Directed Multigraphs*. 2024. arXiv: 2306.11586.

[5] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. *Revisiting Deep Learning Models for Tabular Data*. 2023. arXiv: 2106.11959.

[6] L. Grinsztajn, E. Oyallon, and G. Varoquaux. *Why do tree-based models still outperform deep learning on tabular data?* 2022. arXiv: 2207.08815.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685.

Delft University of Technology