# Tab-AML: A Transformer Based Transaction Monitoring Model for Anti-Money Laundering

Berkan Oztas*, Deniz Cetinkaya*, Festus Adedoyin*, Marcin Budka*, Huseyin Dogan*, Gokhan Aksu†
*Department of Computing and Informatics, Bournemouth University, Bournemouth, United Kingdom
{boztas,dcetinkaya,fadedoyin,mbudka,hdogan}@bournemouth.ac.uk
†Department of Financial Crime Controls, Danske Bank Group, London, United Kingdom
gokhan.aksu@uk.danskebank.com

***Abstract*—Money laundering signifies a major challenge and risk in the global economic landscape. We introduce Tab-AML, a deep-learning model for transaction monitoring that achieves high detection accuracy while significantly reducing false positives. Incorporating a dual-masked Transformer encoder with a shared embedding component and Residual Attention Layers, Tab-AML achieved an ROC-AUC of 93.01, a 98% true positive rate, and a 51% false positive rate on the SAML-D dataset, outperforming models such as TabTransformer and XGBoost. These findings highlight the potential of transformer-based models in advancing anti-money laundering efforts.**

***Index Terms*—Anti-Money Laundering, Artificial Intelligence, Machine Learning, Suspicious Transactions, Transaction Monitoring, Transformer Encoder, Deep Learning**

## I. Introduction

Money laundering poses a significant threat to financial systems, damaging economies and enabling further criminal activities. To combat this, financial institutions are mandated by authorities like the Financial Action Task Force (FATF) to implement anti-money laundering (AML) regimes, including transaction monitoring. However, traditional rule-based approaches dominate this process, relying on manually crafted rules that require frequent updates to adapt to changing behaviours. These methods are highly inefficient, generating over 95% false positives [1], leading to wasted resources and increased operational costs [2].

This paper proposes a deep-learning approach to transaction monitoring, aiming to improve upon existing methods by reducing false positive alerts while maintaining a high true positive rate. By enhancing the accuracy and efficiency of AML systems, our approach seeks to alleviate the burden on human investigators, empowering them to focus their efforts on high-risk cases and make better-informed decisions. Our work focuses on leveraging transformers [3], a largely unexplored approach in this domain, to enhance the effectiveness of transaction monitoring. We present in-depth experiments comparing the performance of our transformer-based model, Tab-AML, against the established TabTransformer model [4], TabNet [5], and various baseline models, highlighting its potential to advance the field.

## II. Related Work

Machine learning techniques have been widely employed to detect money laundering across various domains, with a focus on transaction monitoring in financial institutions. Common approaches include decision trees, random forests, neural networks (NN), and support vector machines (SVM) [6]. Studies have demonstrated the effectiveness of NN-based methods, such as single-layer feedforward networks [7] and radial basis function networks, in enhancing learning speed and accuracy. Random forest models, particularly those leveraging time-frequency attributes, have also shown strong performance in transaction monitoring [8].

Other approaches include isolation forests and one-class SVMs, which perform well with synthetic datasets by flagging anomalous patterns in transaction amounts and timings [9], [10]. XGBoost has also been effective, outperforming rule-based systems and achieving high precision and recall scores [11]. Generative models, such as autoencoders and Wasserstein GANs, have been used to reduce false positives by modelling normal transaction patterns [12].

Emerging methods like graph neural networks (GCNs) have been explored for blockchain transaction analysis, with studies reporting high precision and F1 scores [13]. Transformer-based models, such as HAMLET, have demonstrated their potential in capital market transactions, achieving exceptional classification performance [14].

While these studies highlight the promise of machine learning in AML, gaps persist. Few studies focus on applying deep learning, particularly transformer-based models, to transaction monitoring in financial institutions [15]. Addressing these gaps can advance AML systems by improving detection accuracy and reducing false positives, enabling better identification of illicit activities across financial domains.

As stated in the new Schneiderman book 'Human-Centered AI' [16], researchers, developers, business leaders, policy-makers, and others are expanding the scope of AI centered technology to include the human-centered AI (HCAI) way of thinking, and a bright future awaits AI researchers to include HCAI design and testing strategies when building AI solutions. This paper explains the features of SMAL-D, the architecture of the Tab-AML model, and the experimental setting to provide a level of transparency and traceability in terms of the improvements made to transaction monitoring through the use of a transformer-based deep learning model.

## III. Data

In this work, we utilised the SAML-D dataset [17], a synthetic dataset of 9.5 million transactions, developed using insights from AML specialists through semi-structured interviews, analysis of existing datasets, and a comprehensive literature review. The dataset includes 676,912 unique accounts conducting transactions across 18 geographic locations, using 13 currencies and 7 payment methods. Among these, 0.104% of the transactions (9,873 entries) are labelled as suspicious, representing various money laundering techniques such as layering, rapid fund movement, and transactions to high-risk locations. The dataset includes 28 typologies, 11 normal and 17 suspicious, modelled to reflect real-world scenarios, with overlapping behaviours between normal and suspicious transactions to increase complexity.

TABLE I: SAML-D Features

| Feature | Description |
|---|---|
| **Temporal details** | Date and time stamps of transactions. |
| **Account details** | The account number of the sender and receiver of the transaction. |
| **Transaction Amount** | Indicating the monetary value of each transaction. |
| **Bank Locations** | Geographical data points for the sender and receiver accounts. |
| **Currency Information** | Currency of the transaction for the sender and receiver accounts. |
| **Payment Method** | Types of payments like wire transfer, credit, debit, etc. |
| **Suspicion Indicator** | A binary value signaling normal or suspicious transactions. |
| **Typology Classification** | Identifying the specific AML typology of each transaction. |

Pre-processing involved label encoding of categorical attributes, such as geographical locations, payment types, and currencies, which were numerically encoded for use with embedding layers. Log transformation was applied to the highly skewed Amount feature to normalise its distribution and reduce the impact of outliers, followed by standardisation of non-categorical features to improve model convergence and efficiency. Feature engineering added temporal attributes like Day, Month, Year, Hour, Is Weekend, and Day of Week derived from the Date and Time columns, enabling the model to detect behavioural patterns over time. Finally, unnecessary columns such as Typology Classification and original temporal fields were dropped to streamline the dataset for experimentation. These steps ensured a comprehensive and robust dataset for evaluating transaction monitoring models effectively.

## IV. Model Architecture

This work presents Tab-AML, a transformer-based model designed to enhance transaction monitoring in the AML domain, addressing inefficiencies in rule-based methods and reducing false positives. Tab-AML builds on the TabTransformer [4], incorporating dual-masked transformer encoders with shared embeddings and residual attention mechanisms to capture complex feature interactions and improve detection accuracy. The architecture is illustrated in Figure 1.

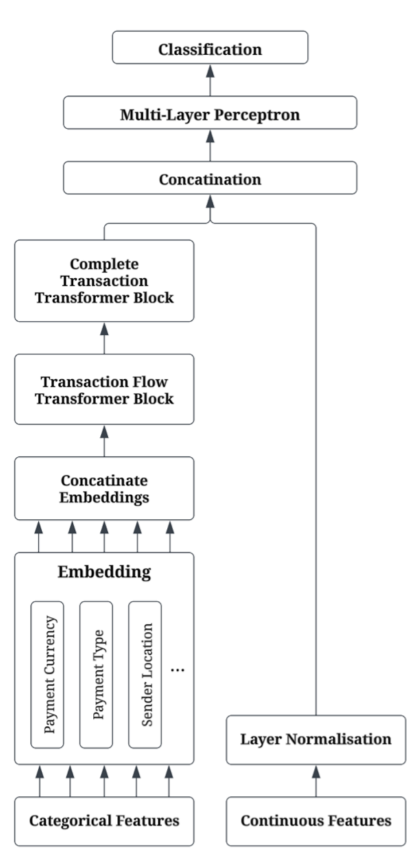



Fig. 1: Overview of the Tab-AML Model Architecture.

### A. Embedding

The embedding stage processes categorical and continuous features, which are common in transactional datasets. Categorical features are embedded using column-based embedding, with dimensions matching the model's dimensions ($d$), adhering to the method used in [4]. The embedding layers are initialised with weights from a truncated normal distribution (mean 0, standard deviation 0.01) to stabilise training and prevent large initial weights from disrupting the learning process. In total 12 features are embedded.

Tab-AML incorporates shared embedding components, where a universal embedding vector is concatenated with individual feature embeddings to improve generalisation across features. Several shared-to-individual embedding ratios (1/4, 1/8, 1/12) were tested, with the 1/8 ratio delivering optimal performance. After embedding, the categorical feature embeddings are concatenated into a single tensor to enable interaction analysis.

### B. Transformers

The concatenated embeddings are fed into dual-masked transformer encoder blocks. The first encoder employs masked attention, exclusively focusing on the 'Sender Account' and 'Receiver Account' features. This selective attention aims to effectively isolate and enhance the interaction analysis between these pivotal features, crucial in contexts where money laundering techniques often involve complex transaction networks designed to hide the illicit origins of funds, such as through structuring [18].

The output from the first encoder is passed to the second encoder, which applies unmasked attention to integrate these focused interactions with the broader dataset context. This two-stage processing aims to capture both granular and global patterns. Residual attention is integrated into the encoder layers to stabilise training, improve context retention, and mitigate overfitting [19]. This mechanism allow the model to identify long-range dependencies and relationships, essential for understanding intricate laundering patterns. A residual dropout is also applied [20].

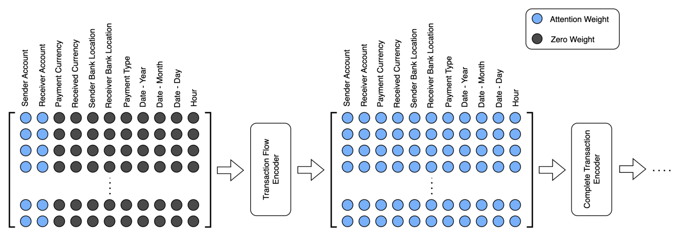



Fig. 2: Dual-masked transformer encoder structure highlighting selective feature processing for money laundering.

### C. Classification

In the final stage of Tab-AML, the contextual embeddings are combined with continuous features to create a vector of dimensions $(d \times \textit{num\_cat}) + \textit{num\_con}$, where $d$ is the embedding dimension, *num_cat* the number of categorical features, and *num_con* the number of continuous features. This vector is then passed to a Multi-Layer Perceptron (MLP) for classification [21].

The MLP architecture uses layer sizes determined by hyperparameters, expanding the input dimension ($l$) by a factor of 4 in the first layer and reducing it by a factor of 2 in the second layer [4]. The final output layer produces a probability using a sigmoid activation function, indicating the likelihood of a transaction being suspicious [22]. Hidden layers employ the Gaussian Error Linear Unit (GELU) activation function for smooth, non-linear transformations, enhancing the model's ability to capture complex patterns in the data [23]. To mitigate overfitting and improve generalisation, dropout is applied within the MLP.

### D. Residual Attention Mechanisms

The attention mechanism in Tab-AML is based on scaled dot-product attention, which computes attention scores using queries ($Q$), keys ($K$), and values ($V$). These scores are scaled by the dimensionality of the key vectors ($d_k$) and normalised using the Softmax function. Multi-head attention extends this mechanism by performing multiple attention operations in parallel, enabling the model to learn diverse subspace representations.

Residual attention enhances the standard attention mechanism, [24], [3], by incorporating attention scores from the previous layer into the current layer's calculations before normalisation. This addition improves the model's ability to capture long-range dependencies and stabilises gradients during training, reducing the risk of overfitting. This modification is shown in Figure 3 and can be described by the following formula:

$$\text{Attention}_{\text{res}}(Q, K, V, \textit{prev}) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \textit{prev}\right) V \quad (1)$$

Where Q, K, and V are the queries, keys, and values, respectively, as in standard attention and *prev* represents the attention scores from the previous layer.

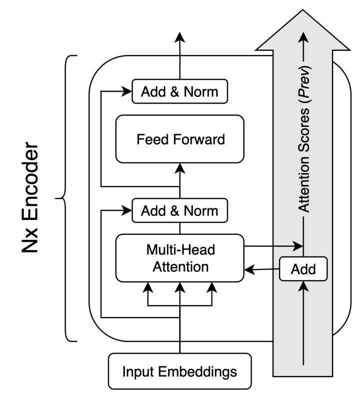



Fig. 3: Transformer Encoder with residual attention used in Tab-AML

## V. Experimental Settings

Our experiments aim to evaluate the performance of the Tab-AML model against frequently employed techniques in money laundering detection, as well as the TabTransformer and TabNet models. Additionally, we optimise the models' performance through hyperparameter tuning. The optimal parameters identified for each model are applied to a holdout dataset to assess their capability in detecting suspicious transactions. Tables V and VI present the chosen hyperparameters and summarise the performance of each model on the validation and test datasets.

### A. Dataset Splitting

We adopt an 80-10-10 time-based train-validation-test split, with transactions chronologically ordered by date and time before splitting. The training dataset comprises 80% of the data (7,603,881 transactions) and is used to fit the models. The validation and test datasets share the remaining 20% equally, each containing 950,485 transactions. The validation dataset is employed for hyperparameter tuning, while the test dataset evaluates the models' generalisability. To prevent potential bias, we avoided retraining the models on the combined training and validation datasets after hyperparameter tuning.

## B. Evaluation Metrics

Given the highly imbalanced nature of the dataset, we use the ROC-AUC score [25] as the primary evaluation metric, as traditional accuracy metrics are unsuitable. The ROC-AUC score assesses the model's ability to distinguish between normal and suspicious transactions by plotting the true positive rate against the false positive rate across varying thresholds. To further analyse the best-performing model, we generate the ROC curve and confusion matrix at a specific threshold, providing insight into the model's true and false positive rates.

## C. Hyperparameter Tuning: Deep Learning Models

This section outlines our hyperparameter tuning strategy, detailing the exploration of specific value ranges. Default values were used for any hyperparameters not explicitly discussed. Due to computational constraints, we opted for a manual hyperparameter strategy instead of more resource-intensive methods like GridSearch or Optuna [26]. This approach involved systematically testing one hyperparameter at a time, keeping others constant to isolate the effects of individual hyperparameters. Our efforts aimed to maximise the AUC-ROC score on the validation dataset. Despite the limitations of manual tuning, such as time intensity and the potential for incomplete exploration, this method yielded valuable insights into hyperparameter sensitivity and model behaviour. Early stopping was implemented to stop training for Tab-AML and TabTransformer when score improvement fell below $1e^{-4}$, while TabNet halted after five epochs without AUC-ROC improvement.

Tables II and III highlight the tested hyperparameters for each model, outlining the ranges and configurations systematically explored during the tuning process.

For the Tab-AML and TabTransformer models, hyperparameters were tuned based on guidelines from [4] and [19]. Binary Cross-Entropy with Logits Loss (BCEwithLogitLoss) was chosen for its stability in classification tasks. Dropout rates of 0%, 10%, and 20% were tested across different components, with 10% proving optimal for all, reducing overfitting while preserving learning capacity.

TABLE II: Tested Hyperparameters for the Tab-AML and TabTransformer Models

| Parameter | Tested Values |
|---|---|
| Learning Rate ($lr$) | $1e^{-5}$, $1e^{-4}$, $1e^{-3}$, $1e^{-2}$ |
| Optimisers ($O$) | Adam, AdamW, SGD |
| Model Dimensionality ($d$) | 16, 32, 64, 128 |
| Batch Size ($B$) | 32, 64, 128, 256 |
| Multi-Head Attention Configuration | 4 heads with 2 layers, 8 heads with 4 layers |
| Residual/Attention Dropout | 0%, 10%, 20% |
| MLP/Feed Forward Dropout | 0%, 10%, 20% |

The TabNet model's hyperparameters were tuned following the recommendations in [5]. A combination of parameters, including learning rates, decision step depths, attention widths, and batch sizes, were systematically varied to optimise performance. Table III provides an overview of the tested hyperparameter configurations for TabNet.

TABLE III: Tested Hyperparameters for the TabNet Model

| Parameter | Tested Values |
|---|---|
| Learning Rate ($lr$) | $1e^{-3}$, $1e^{-2}$, $1e^{-1}$ |
| Depth of Decision Steps ($N_d$) | 16, 32, 64 |
| Width of Attention Layers ($N_a$) | 16, 32, 64 |
| Number of Decision Steps ($N_{\text{steps}}$) | 1, 3, 5 |
| Feature Scaling ($\gamma$) | 1.0, 1.5, 2.0 |
| Sparse Regularisation ($\lambda_{\text{sparse}}$) | $1e^{-6}$, $1e^{-4}$, $1e^{-2}$ |
| Batch Size ($B$) | 2048, 4096, 16384 |
| Virtual Batch Size ($VB$) | 512, 1024, 2048 |
| Momentum ($M$) | 0.1, 0.25, 0.4 |

## D. Hyperparameter Tuning: Baseline Models

Baseline models were optimised using GridSearchCV [27] with 3-fold cross-validation, focusing on ROC-AUC scores. These models included logistic regression, K-Nearest Neighbours (KNN), decision trees, random forests, XGBoost, and Gaussian Naive Bayes, implemented via Scikit-learn [28] and XGBoost libraries.

TABLE IV: Tested Hyperparameters for Baseline Models

| Model | Hyperparameters |
|---|---|
| Logistic Regression | Solver: lbfgs; $C \in \{0.01, 0.1, 1, 10, 100\}$ |
| K-Nearest Neighbours (KNN) | Number of Neighbours: $N \in \{2, 3, 5, 7, 9\}$ |
| Decision Trees | Max Depth: $D \in \{2, 4, 8, 16\}$; Min Samples Split: $S \in \{4, 8, 12\}$ |
| Random Forest | Max Depth: $D \in \{2, 4, 8, 12\}$; Max Features: $F \in \{2, 4, 8, 12\}$ |
| XGBoost | Max Depth: $D \in \{2, 4, 8, 16\}$; Learning Rate: $L \in \{0.1, 0.2, 0.3\}$ |
| Gaussian Naive Bayes (GaussianNB) | No hyperparameter tuning |

# VI. Results

Once the optimal parameters were identified using the validation dataset, the models were retrained on the test dataset to benchmark their performance. Tables V and VI highlight the effectiveness of the Tab-AML model in detecting suspicious transactions compared to other methods. Tab-AML achieved an ROC-AUC score of 92.83% on the validation dataset and 93.01% on the test dataset, outperforming all other approaches. The TabTransformer achieved an ROC-AUC of 87.66% on the validation dataset and 85.94% on the test dataset, outperforming TabNet and baseline models. TabNet obtained ROC-AUC scores of 81.78% (validation) and 80.61% (test). Among traditional machine learning approaches, XGBoost performed the best, aligning with existing literature highlighting its effectiveness [29].

The superior performance of Tab-AML can be attributed to its architectural improvements, including the dual-masked Transformer encoder, shared embeddings, and residual attention mechanism, which effectively capture both micro and macro patterns. The shared embedding mechanism enables better learning of feature interactions, improving the model's handling of complex transactional data. The first transformer block focuses on local interactions, while the second block

TABLE V: ROC Scores and Best Hyperparameters for the Models on the Validation Dataset. Displays the average ROC scores, computed over three different seeds, alongside the optimal hyperparameters identified for each model.

| **Models** | **ROC-AUC Score** | **Hyperparameters** |
|---|---|---|
| Tab-AML | **92.83** (0.13) | $O$ = AdamW, $lr$ = $1e^{-4}$, $d = 32$, $B = 64$, Heads = 4, Layers = 2 |
| TabTransformer | 87.66 (0.69) | $O$ = AdamW,$lr$= $1e^{-3}$, $d = 64$, $B = 128$, Heads = 4, Layers = 2 |
| TabNet | 81.78 (0.01) | $O$ = Adam, $lr$= $2e^{-2}$, $N_d = N_a = 32$, $N_{\text{steps}} = 1$, $\gamma = 2$, $\lambda_{\text{sparse}} = 1e^{-6}$, $M = 0.4$ , $B = 4096$, $VB = 512$ |
| **Baseline Models** | | |
| Logistic Regression | 59.74 | $C$ = 100 |
| K-Nearest Neighbours | 59.09 | $N$ = 7 |
| Decision Tree | 78.44 | $D$ = 8, $S$ = 4 |
| Random Forest | 80.51 | $D$ = 8, $F$ = 8 |
| XGBoost | 81.64 | $D$ = 4, $L$ = 0.1 |
| Gaussian Naive Bayes | 66.55 | - |

TABLE VI: ROC Scores of the Models on the Test Dataset. Displays the average ROC scores for each model, computed over three different seeds.

| **Models** | **ROC-AUC Score** |
|---|---|
| Tab-AML | **93.01** (0.61) |
| TabTransformer | 85.94 (0.24) |
| TabNet | 80.61 (0.03) |
| **Baseline Models** | |
| Logistic Regression | 56.59 |
| K-Nearest Neighbours | 59.61 |
| Decision Tree | 77.85 |
| Random Forest | 79.69 |
| XGBoost | 81.12 |
| Gaussian Naive Bayes | 63.89 |

integrates global feature patterns. The residual attention mechanism further enhances long-range dependency modelling, crucial for detecting money laundering behaviours over time [19].

Figures 4(a) and 4(b) show the ROC curves for Tab-AML and TabTransformer. Tab-AML consistently outperformed TabTransformer across thresholds. At a threshold producing equal true positive rates, Tab-AML achieved a false positive rate of 51%, compared to TabTransformer's 68% (Figures 4(c) and 4(d)). These results confirm Tab-AML's superior ability to identify complex patterns in suspicious transactions.

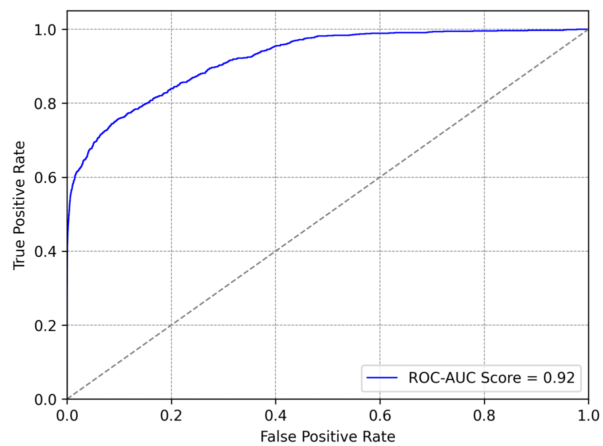
## VII. Conclusion and Future Work

This study introduces Tab-AML, a transformer-based deep learning model for detecting potential money laundering transactions. By leveraging shared embeddings and a dual-masked transformer encoder with residual attention, the model effectively captures complex transaction patterns. The first transformer block focuses on micro-level interactions (e.g., sender-receiver flows), while the second examines macro-level feature sets. Tab-AML achieved an ROC-AUC score of 93.01%, reducing the false positive rate by 17% compared to TabTransformer (85.94%) and significantly surpassing XGBoost (81.12%) and TabNet (80.61%). This achievement is aimed at augmenting human decision-making in anti-money laundering (AML) systems and directly addresses the need for systems that enhance human decision making and reduce investigator fatigue.

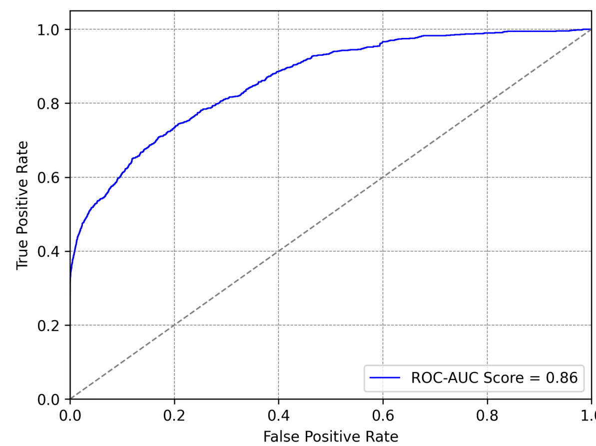
Experiments were conducted on the SAML-D synthetic dataset, developed with AML specialists. Future work involves validating Tab-AML on real datasets, exploring feature engineering, and enhancing explainability through attention mechanisms to foster trustworthiness. Integrating Large Language Models (LLMs) could further refine investigations, leveraging explainability tools to streamline alert analysis, ensuring fairness, and improved human-AI collaboration.

Tab-AML offers financial institutions improved detection efficiency and lower operational costs. It can integrate with existing rule-based systems as an additional layer, enhancing flagged transaction analysis while preserving human oversight and situational awareness in decision-making. This research highlights the potential of transformer models in AML and aligns with the goal of building responsible, human-compatible AI solutions that augment human intelligence and trust in technology.

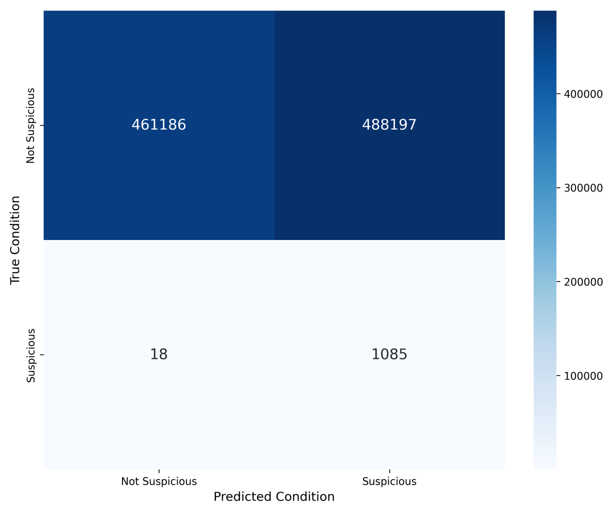
AML detection systems need to be human-centric, situating explainable AI across the financial domains and helping stakeholders combat financial crime more effectively. There is a need to develop human-centered AML detection systems that require minimal training for end-users.



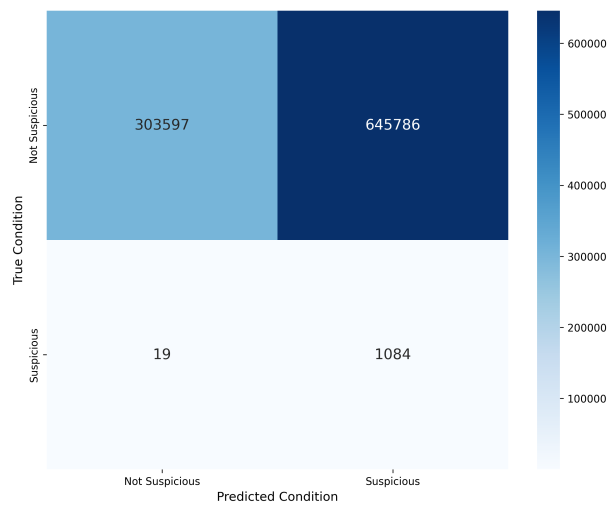

(a) Tab-AML model's ROC curve.




(b) TabTransformer model's ROC curve.




(c) Tab-AML confusion matrix: True Positive Rate: 98%, False Positive Rate: 51%.




(d) TabTransformer confusion matrix: True Positive Rate: 98%, False Positive Rate: 68%.

Fig. 4: Comparative analysis of ROC curves and confusion matrices for the Tab-AML and TabTransformer models.

## REFERENCES

[1] E. Eifrem, "How graph technology can map patterns to mitigate money laundering risk," Computer Fraud & Security, vol. 2019, no. 10, pp. 6–8, 2019. doi:10.1016/S1361-3723(19)30105-8.

[2] B. Oztas, D. Cetinkaya, F. Adedoyin, M. Budka, H. Dogan, and G. Aksu, "Perspectives from Experts on Developing Transaction Monitoring Methods for Anti-Money Laundering," 2023 IEEE International Conference on e-Business Engineering (ICEBE), Sydney, Australia, pp. 39–46, 2023. doi:10.1109/ICEBE59045.2023.00024.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, pp. 6000–6010, 2017. doi:10.48550/arXiv.1706.03762.

[4] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular Data Modeling Using Contextual Embeddings," arXiv preprint arXiv:2012.06678, 2020. doi:10.48550/arXiv.2012.06678.

[5] S. "O. Arik and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 8, pp. 6679–6687, 2021. doi:10.1609/aaai.v35i8.16826.

[6] Y. Zhang and P. Trubey, "Machine learning and sampling scheme: An empirical study of money laundering detection," Computational Economics, vol. 54, pp. 1043–1063, 2019. doi:10.1007/s10614-018-9864-z.

[7] L. T. Lv, N. Ji, and J. L. Zhang, "A RBF neural network model for anti-money laundering," 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, China, pp. 209–215, 2008. doi:10.1109/ICWAPR.2008.4635778.

[8] U. G. Ketenci, T. Kurt, S. Onal, C. Erbil, S. Akturkoglu, and H. S. Ilhan, "A Time-Frequency Based Suspicious Activity Detection for Anti-Money Laundering," IEEE Access, vol. 9, pp. 59957–59967, 2021. doi:10.1109/ACCESS.2021.3072114.

[9] A. E. M. Shokry, M. A. Rizka, and N. M. Labib, "Counter terrorism finance by detecting money laundering hidden networks using unsupervised machine learning algorithm," International Conference on e-Learning, pp. 89–97, 2020. doi:10.33965/ict_csc_wbc_2020_202008l012.

[10] J. Guevara, O. Garcia-Bedoya, and O. Granados, "Machine learning methodologies against money laundering in non-banking correspondents," Applied Informatics. ICAI 2020. Communications in Computer and Information Science, vol. 1277, pp. 72–88, 2020. doi:10.1007/978-3-030-61702-8_6.

[11] M. Jullum, A. Løland, R. B. Huseby, G. Ånonsen, and J. Lorentzen, "Detecting money laundering transactions with machine learning," Journal of Money Laundering Control, vol. 23, no. 1, pp. 173–186, 2020.

doi:10.1108/JMLC-07-2019-0055.

[12] Z. Chen, W. M. Soliman, N. Amril, and S. Mohammad, ”Variational Autoencoders and Wasserstein Generative Adversarial Networks for Improving the Anti-Money Laundering Process,” IEEE Access, vol. 9, pp. 83762–83785, 2021. doi:10.1109/ACCESS.2021.3086359.

[13] I. Alarab, S. Prakoonwit, and M. I. Nacer, ”Competence of Graph Convolutional Networks for Anti-Money Laundering in Bitcoin Blockchain,” Proceedings of the 2020 5th International Conference on Machine Learning Technologies, Beijing, China, pp. 23–27, 2020. doi:10.1145/3409073.3409080.

[14] M. P. Tatulli, T. Paladini, M. D’Onghia, M. Carminati, and S. Zanero, ”HAMLET: A Transformer Based Approach for Money Laundering Detection,” Cyber Security, Cryptology, and Machine Learning: 7th International Symposium, CSCML 2023, Be’er Sheva, Israel, Proceedings, pp. 234–250, 2023. doi:10.1007/978-3-031-34671-2_17.

[15] D. V. Kute, B. Pradhan, N. Shukla, and A. Alamri, ”Deep Learning and Explainable Artificial Intelligence Techniques Applied for Detecting Money Laundering–A Critical Review,” IEEE Access, vol. 9, pp. 82300–82317, 2021. doi:10.1109/ACCESS.2021.3086230.

[16] B. Shneiderman, Human-Centered AI, Oxford University Press, 2022.

[17] B. Oztas, D. Cetinkaya, F. Adedoyin, M. Budka, H. Dogan, and G. Aksu, ”Enhancing Anti-Money Laundering: Development of a Synthetic Transaction Monitoring Dataset,” 2023 IEEE International Conference on e-Business Engineering (ICEBE), Sydney, Australia, pp. 47–54, 2023. doi:10.1109/ICEBE59045.2023.00028.

[18] O. M. Granados and A. Vargas, ”The geometry of suspicious money laundering activities in financial networks,” EPJ Data Science, vol. 11, no. 6, 2022. doi:10.1140/epjds/s13688-022-00318-w.

[19] R. He, A. Ravula, B. Kanagal, and J. Ainslie, ”RealFormer: Transformer Likes Residual Attention,” arXiv preprint arXiv:2012.11747, 2021. doi:10.48550/arXiv.2012.11747.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ”Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014.

[21] H. Taud and J.-F. Mas, ”Multilayer Perceptron (MLP),” Geomatic Approaches for Modeling Land Change Scenarios, Springer, Cham, pp. Chapter 27, 2018. doi:10.1007/978-3-319-60801-3_27.

[22] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, ”Efficient backprop,” Neural Networks: Tricks of the Trade, Springer Verlag, Lecture Notes in Computer Science, vol. 7700, pp. 9–48, 2012. doi:10.1007/978-3-642-35289-8_3.

[23] D. Hendrycks and K. Gimpel, ”Gaussian Error Linear Units (GELUs),” arXiv preprint arXiv:1606.08415, 2016. doi:10.48550/arXiv.1606.08415.

[24] M.-T. Luong, H. Pham, and C. D. Manning, ”Effective Approaches to Attention-based Neural Machine Translation,” arXiv preprint arXiv:1508.04025, 2015. doi:10.48550/arXiv.1508.04025.

[25] T. Fawcett, ”An introduction to ROC analysis,” Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006. doi:10.1016/j.patrec.2005.10.010.

[26] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, ”Optuna: A Next-Generation Hyperparameter Optimization Framework,” The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631, 2019.

[27] P. Liashchynskyi and P. Liashchynskyi, ”Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS,” 2019. arXiv:1912.06059. doi:10.48550/arXiv.1912.06059.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., ”Scikit-learn: Machine Learning in Python,” arXiv preprint arXiv:1201.0490, 2018. doi:10.48550/arXiv.1201.0490.

[29] J. Domashova and N. Mikhailina, ”Usage of machine learning methods for early detection of money laundering schemes,” Procedia Computer Science, vol. 190, pp. 184–192, 2021. doi:10.1016/j.procs.2021.06.033.