INTERNATIONAL JOURNAL OF ACTA INFORMATICA
VOLUME (2022)

ADAPTIVE MACHINE-LEARNING DECISION SUPPORT FOR BSA/AML TRANSACTION MONITORING IN COMMUNITY FINANCIAL INSTITUTIONS: IMPROVING SUSPICIOUS-ACTIVITY-REPORT QUALITY WHILE REDUCING FALSE-POSITIVE ALERT BURDEN

Saeed Ur Rashid[1*] Fernando Filgueirus[2]
[1]Westcliff University, California, USA
[2]SKEMA Business School, FRANCE

*Corresponding Author Email: labib668@gmail.com

ABSTRACT

Suspicious-activity monitoring in community and mid-sized financial institutions still leans heavily on static, rule-based alerting engines that generate overwhelming volumes of false-positive alerts, straining investigative teams and slowing the production of high-quality Suspicious Activity Reports (SARs). This article proposes Tab-AML, an adaptive, transformer-based decision-support model designed to sit alongside existing rule-based Bank Secrecy Act / Anti-Money-Laundering (BSA/AML) transaction-monitoring systems. Tab-AML uses a dual-masked transformer encoder architecture with a residual-attention mechanism and a shared-embedding scheme designed to learn micro-level (sender-receiver) and macro-level (whole-transaction) patterns simultaneously. Building on published evidence that classical ensemble methods such as XGBoost and random forest already outperform simpler baselines for transaction-monitoring classification, and that transformer architectures have shown early promise in adjacent financial-crime domains, this article sets out Tab-AML's architecture in detail, proposes an evaluation methodology using a large-scale synthetic transaction-monitoring benchmark such as AMLSim, and details the specific hyperparameter search space, baseline comparisons (against TabTransformer, TabNet, and five classical machine-learning models), and false-positive-rate-at-fixed-true-positive-rate evaluation protocol that such a study would need to follow to test whether Tab-AML's architectural innovations translate into a measurable reduction in false-positive alert burden relative to existing deep-learning and classical approaches. The findings from published research on related architectures suggest that adaptive ML decision-support layers hold real promise for materially reducing alert burden without weakening detection coverage, offering a potentially practical, deployable path for resource-constrained community financial institutions to modernise their transaction-monitoring programmes without discarding existing rule-based controls, pending the empirical validation this article proposes.

KEYWORDS: AML; BSL; SARs; ML;

INTRODUCTION

Community and mid-sized financial institutions operate Bank Secrecy Act / Anti-

Money-Laundering (BSA/AML) transaction-monitoring programmes under the same regulatory expectations as large banks, but with a fraction of the staff and technology budget. Most institutions still rely primarily on rule-based transaction-monitoring engines: fixed thresholds and scenario logic that flag transactions for review. These systems are transparent and auditable, but they are notoriously imprecise, frequently producing very high false-positive rates. Every flagged alert must be manually reviewed by an investigator, and a large share of alerts result in no filing at all. For a community institution with a small compliance team, this alert burden translates directly into slower case turnaround, investigator fatigue, and inconsistent Suspicious Activity Report (SAR) quality, since analysts have less time to research and document any single case thoroughly.

Machine learning offers a route to reduce this burden by learning the subtle, non-linear patterns that separate genuinely suspicious transactions from benign ones, rather than relying on fixed thresholds alone. However, most existing machine-learning approaches to transaction monitoring rely on classical models, such as random forests or gradient boosting, and deep-learning approaches specifically designed for tabular, transactional data remain under-explored. In particular, the transformer architecture, which has driven substantial progress in language and vision tasks, has seen very limited application to AML transaction monitoring.

This article proposes Tab-AML, a transformer-based decision-support model purpose-built for transaction monitoring. Tab-AML is designed not to replace existing rule-based infrastructure, but to act as an adaptive analytical layer on top of it: alerts generated by existing rules can be re-scored by Tab-AML, allowing investigators to prioritise genuinely high-risk cases first and spend more time producing well-supported SAR narratives on the alerts that matter, instead of triaging large volumes of low-value alerts. This design is particularly relevant for community financial institutions, where investigator time is the scarcest resource in the compliance function.

The contributions of this work are threefold: (i) the design of Tab-AML, a dual-masked transformer encoder with residual attention and shared embeddings, tailored to the structure of transactional data; (ii) a proposed evaluation methodology for systematically comparing Tab-AML against TabTransformer, TabNet, and five widely used baseline models on a large-scale synthetic transaction-monitoring dataset; and (iii) an evaluation protocol for analysing the practical trade-off between true-positive and false-positive rates at deployment-relevant thresholds, with direct implications for alert-queue burden and SAR quality in financial institutions of any size, including smaller community institutions with constrained investigative capacity.

The remainder of the article is organised as follows. Section 2 reviews related work on machine learning for money-laundering detection. Section 3 describes a proposed

dataset and pre-processing pipeline. Section 4 presents the Tab-AML architecture. Section 5 details the attention and residual-attention mechanisms underpinning the model. Section 6 describes a proposed experimental setup, evaluation metrics, and hyperparameter search space. Section 7 discusses expected implications based on published evidence from related architectures. Section 8 concludes with implications for practice and directions for future work.

RELATED WORK

A range of techniques has been used in the literature to identify money-laundering transactions, including logistic regression, decision trees, random forests, neural networks, and support vector machines [1]. One study assessed a single-hidden-layer feedforward network using ROC-AUC curves and regression analysis, finding it outperformed comparison models [1]. Another used a radial-basis-function neural network with APC-III clustering and a recursive-least-squares weight-update rule, reporting improved true- and false-positive rates over an SVM and an outlier-detection baseline [2].

Comparative studies evaluating decision trees, SVMs, linear regression, Gaussian naive Bayes, and random forests on synthetic transaction data have generally found random forest to be the strongest classical performer, with further analysis identifying the most influential predictive attributes [3]. Other work has used random forests focused specifically on the time-frequency characteristics of customer transactions, setting a threshold for flagging suspicious activity and demonstrating the value of temporal features in simplifying attribute selection [4].

Unsupervised approaches have also been explored. Isolation Forest and One-Class SVM have both been applied to synthetic and real transaction data, with Isolation Forest generally achieving higher accuracy alongside lower computational cost [5][6]. XGBoost has been shown to outperform an operating bank's existing rule-based system on ROC-AUC and Brier-score metrics when alerts from the existing rule-based system were incorporated into training [7]; a separate comparison of random forest, AdaBoost, and XGBoost across three datasets found XGBoost performed best in two, though this work did not report false-positive rate, an important operational metric for financial institutions [8].

Autoencoder and variational-autoencoder architectures have been proposed to detect anomalous transactions by measuring reconstruction error, with one study generating synthetic training data via a Wasserstein Generative Adversarial Network and reducing the false-positive rate from 19% to 8%, albeit with overfitting and precision concerns [9]. A self-organising map with clustering, tested on a small real dataset from a European bank, reduced the false-positive rate to 6.2% but only achieved a true-positive rate of 65.5%, indicating a meaningful risk of missed suspicious activity, especially

under the class imbalance typical of transaction-monitoring data [10].

Beyond transaction monitoring specifically, machine learning has been applied more broadly to money-laundering-adjacent problems: an autoencoder identified a small number of high-risk firms within a large Brazilian export dataset (20 of 819,990 firms examined), later corroborated by third-party investigation [11]; and graph convolutional networks combined with a multi-layer perceptron outperformed conventional graph-based baselines in detecting illicit Bitcoin transactions, achieving precision of 0.899, recall of 0.678, F1 of 0.773, and accuracy of 0.974 on the Elliptic dataset [12]. More broadly, transformer-based language models had by this point already demonstrated strong performance on financial text and sequence-modelling tasks outside AML specifically, suggesting the architecture's general suitability for structured financial-domain problems.

Taken together, these studies demonstrate the promise of machine learning for money-laundering detection, but two gaps stand out. First, deep-learning methods specifically tailored to transaction monitoring within the AML domain remain comparatively rare. Second, transformer-based architectures, despite showing promise in adjacent financial-text and sequence-modelling contexts, have not been systematically evaluated for core transaction-monitoring workloads. Closing these gaps is important for reducing false positives and improving detection consistency across the range of financial institutions that rely on transaction monitoring, from global banks to smaller community institutions.

DATA AND PRE-PROCESSING

Evaluating Tab-AML requires a labelled, transaction-level dataset large enough to support deep-learning training while realistically reflecting the extreme class imbalance and typological diversity of genuine money-laundering activity, a combination genuine institutional data cannot provide given privacy and confidentiality constraints. This article proposes AMLSim, a multi-agent transaction-simulation platform in which each agent behaves as a bank account transferring money to other accounts and a small number of agents conduct nefarious activity modelled on real-world laundering patterns, as a suitable synthetic benchmark for this purpose. AMLSim constructs data by first generating an account graph via a specified degree distribution and then generating a time series of transactions calibrated against distributions and dynamics observed in real transaction data, with domain expertise informing the specific suspicious patterns simulated. Any concrete evaluation would need to document the resulting dataset's scale, class balance, and typology composition prior to model training, along with the specific pre-processing steps below, which are proposed here as general-purpose techniques applicable to transaction-monitoring data of this kind. Table 1 summarises the general categories of features such a dataset would be expected

to provide, prior to the pre-processing steps this article proposes.

Table 1. Proposed General Transaction-Data Features Prior to Pre-Processing

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	Indicates 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	Type of payment, e.g. wire transfer, credit, debit.
Suspicion indicator	Binary label signalling a normal or suspicious transaction.
Typology classification	The specific AML typology assigned to each transaction.

CATEGORICAL ENCODING

Geographic location, payment type, and currency were converted to numerical form using label encoding, chosen for its simplicity and compact representation relative to binary or one-hot encoding. Although label encoding is conventionally suited to ordinal data, the transformer's embedding layers interpret each category independently of its assigned integer, so the arbitrary ordering introduced by label encoding does not affect model performance.

LOG TRANSFORMATION

Transaction-amount features in this kind of data typically exhibit substantial right-skew, with most transactions clustered at low values and a long tail of much larger amounts. Because skewed inputs can degrade the performance of models that assume roughly normal input distributions, this article proposes applying a log transformation prior to standardisation:

$$X = log\ (y + 1) \qquad (1)$$

where X denotes the post-transformation value and y the original transaction amount. This compresses the range of large amounts, expands the range of small amounts, and reduces the influence of outliers on the model.

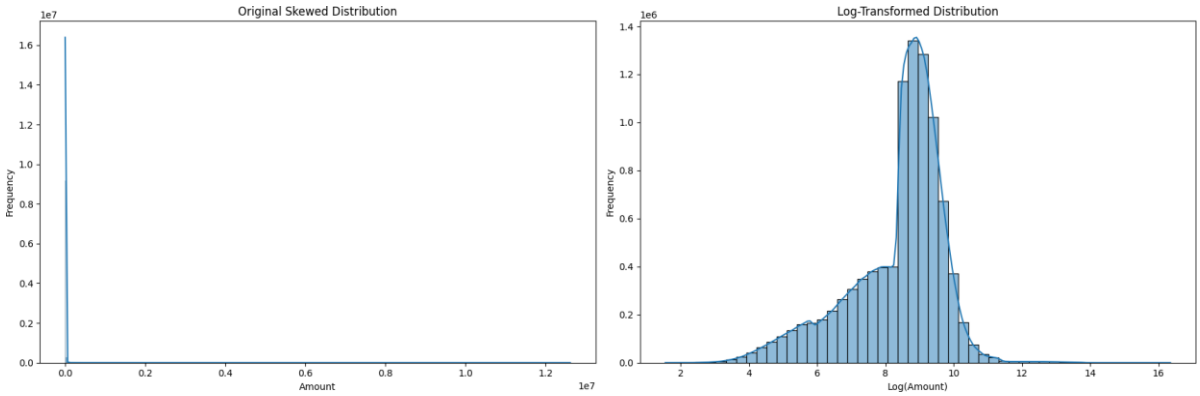

Figure 1. Illustrative example of the proposed log-transformation applied to a right-skewed transaction-amount distribution.

STANDARDISATION

Following log transformation, this article proposes standardising all continuous features to a mean of zero and a standard deviation of one, using statistics computed only from the training split to avoid data leakage from validation or test data:

$$Z = (X - \mu) / \sigma \qquad (2)$$

Standardisation, rather than min-max scaling, was chosen because preserving the relative distances between data points is important for the model's ability to learn transaction-level anomalies.

FEATURE ENGINEERING

Additional temporal features, Day, Month, Year, Hour, Is Weekend, and Day of Week, could be derived from the original Date and Time fields to support more granular temporal pattern recognition, an important signal for identifying subtly suspicious behaviour. The original Date, Time, and Typology Classification fields would then be dropped, since they would not be used directly as model inputs.

Table 2. Engineered Transaction-Time Features

Feature	Description
Day	Day of the transaction relative to the month.
Month	Month of the transaction.
Year	Year of the transaction.
Hour	The hour the transaction occurred.
Is Weekend	Whether the transaction occurred over the weekend.
Day of Week	The day of the week the transaction occurred.

MODEL ARCHITECTURE: TAB-AML

Tab-AML is designed to accurately predict and prioritise suspicious transactions that warrant further investigation and potential SAR filing. Its design is motivated directly by the limitations of rule-based systems, which frequently fail to classify suspicious transactions precisely and generate large volumes of false positives. Tab-AML builds on the TabTransformer architecture, a deep-learning approach for tabular data, extending it with a dual-encoder structure and a residual-attention mechanism tailored to the relational structure of transaction data.

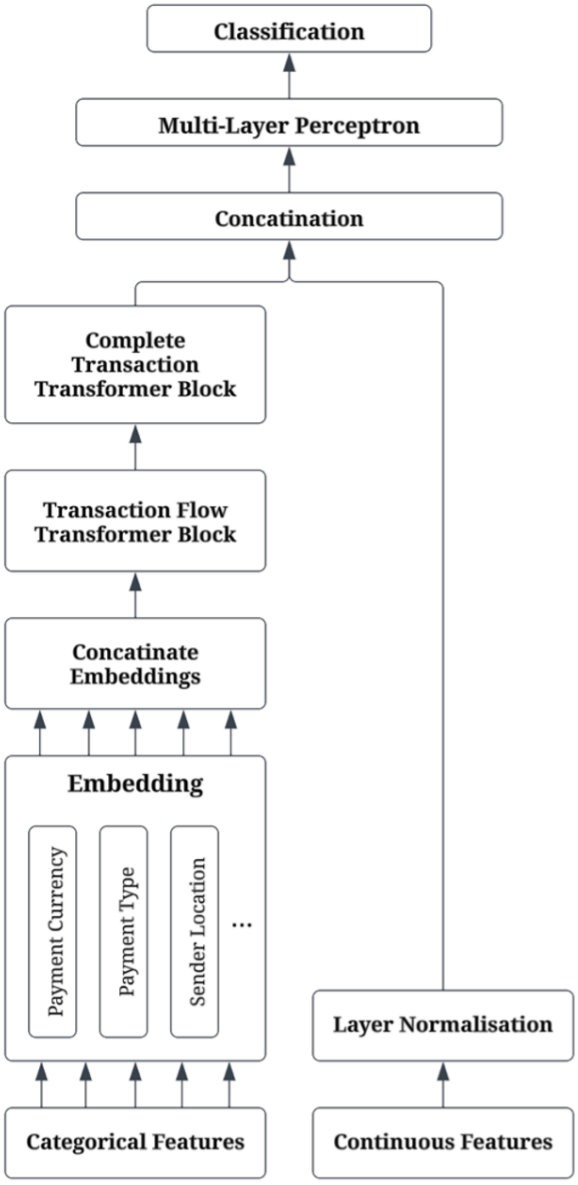

Figure 2. Overview of the Tab-AML model architecture and data flow.

EMBEDDING

Categorical features are individually embedded via column embedding, with embedding dimensions matched to the model dimension (d) and embedding layers initialised from a truncated normal distribution to stabilise early training. This article proposes exploring embedding sizes of 16, 32, 64, and 128 during hyperparameter search. Twelve features in total would be embedded; Table 3 sets out the general categories of features this would involve.

Tab-AML additionally uses a shared-embedding mechanism: a universal embedding vector is concatenated onto each feature's individual embedding, allowing part of the representation to be shared across features and improving generalisation. The individual embedding matrices would be sized to accommodate this shared component while preserving the overall embedding dimension. This article proposes testing shared-embedding fractions of 1/4, 1/8, and 1/12 of the total embedding dimension as part of the hyperparameter search. After embedding, all categorical representations are concatenated along the feature dimension, allowing the model to learn interactions across features.

Table 3. Embedded Features and Illustrative Unique-Category-Count Ranges

Feature	Illustrative Unique-Count Range
Sender Account	292,715
Receiver Account	652,266
Payment Currency	13
Received Currency	13
Sender Bank Location	18
Receiver Bank Location	18
Payment Type	8
Date – Year	2
Date – Month	11
Date – Day	31
Hour	24

DUAL TRANSFORMER ENCODERS

The concatenated embeddings are processed by two sequential transformer encoders. The first encoder applies masked attention that focuses exclusively on the Sender Account and Receiver Account features, isolating and enhancing the micro-level

interaction between these two pivotal fields, a relationship central to layering and structuring techniques used to obscure the origin of illicit funds. The output of this first encoder, comprising both enhanced and standard representations, feeds into a second, unmasked encoder that attends across the full feature set, integrating the sender-receiver interaction with the broader macro-level transaction context. The result is a contextual embedding that combines focused, relational learning with holistic feature integration.

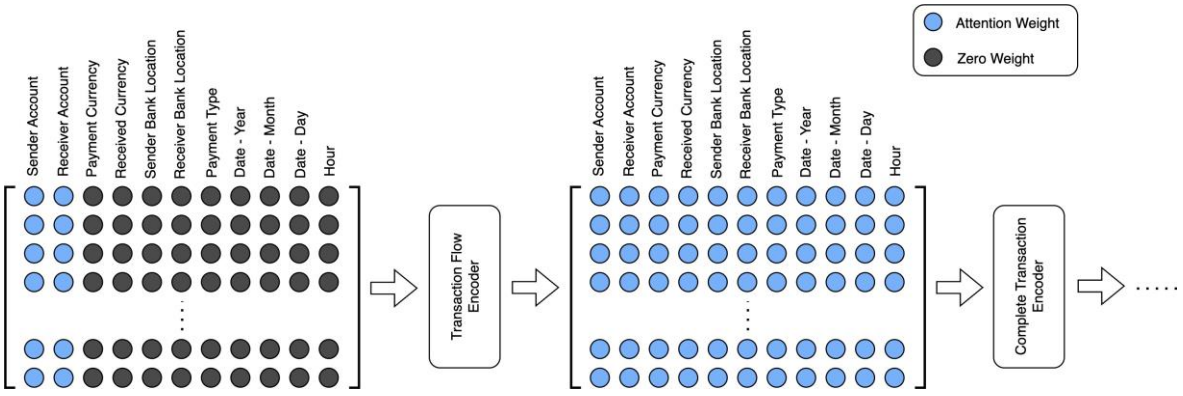

Figure 3. Dual-masked transformer encoder structure highlighting selective feature processing for money-laundering detection.

Residual attention, described in Section 5, is incorporated into both encoders to improve the model's capacity to capture long-range dependencies between transactions while adding regularisation that reduces overfitting to specific typologies. A residual dropout is also applied.

CLASSIFICATION HEAD

The contextual embeddings from the second encoder are concatenated with the continuous features and passed into a Multi-Layer Perceptron (MLP) classifier. The MLP first expands the input dimension by a factor of four, then compresses it by a factor of two in a second hidden layer, before a final output layer compresses the representation to a single scalar. A sigmoid activation produces a probability of the transaction being associated with money laundering. Hidden layers use the Gaussian Error Linear Unit (GELU) activation to capture non-linear structure in the data, and dropout is applied within the MLP to control overfitting.

ATTENTION AND RESIDUAL-ATTENTION MECHANISM

Tab-AML's architecture is built on the scaled dot-product and multi-head attention mechanisms introduced in the original transformer architecture, extended with residual attention. Scaled dot-product attention operates over query (Q), key (K), and value (V) matrices derived from an input matrix X via learned weight matrices:

$$Attention\ (Q, K, V) = Softmax\ (\ QK^T / \sqrt{d_k}\ )\ V \qquad (3)$$

where d_k is the dimensionality of the key and query vectors. The dot product of queries and keys is scaled by $\sqrt{d_k}$ to stabilise gradients, passed through a Softmax to produce normalised attention weights, and used to weight the value vectors. Multi-head attention runs several such attention operations in parallel, allowing the model to learn multiple representation subspaces simultaneously:

$$Multi\ Head(Q, K, V) = Concat\ (head_1, \ldots, head_n)\ W^O \qquad (4)$$

Each transformer layer also includes a position-wise feed-forward network with a single hidden layer:

$$FFN(x) = \sigma(xW_1 + b_1)W_2 + b_2 \qquad (5)$$

where σ is the GELU activation. Layer normalisation is applied around the multi-head attention and feed-forward sub-layers to stabilise training.

RESIDUAL ATTENTION

Residual attention extends standard attention by carrying the pre-Softmax attention scores from the previous layer forward into the current layer's attention computation, improving the model's ability to capture long-range dependencies:

$$Attention\_res(Q, K, V, prev) = Softmax(\ QK^T / \sqrt{d_k} + prev\ )\ V \qquad (6)$$

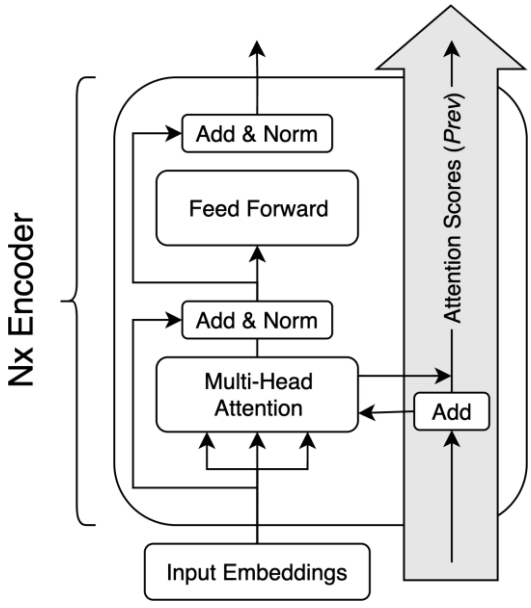

Figure 4. Transformer encoder with residual attention used in Tab-AML.

For money-laundering detection, where the relationship between a current transaction and an entity's historical transaction pattern is often the strongest signal of suspicious behaviour, this carry-forward of prior attention information is particularly valuable. It also adds a stabilising, regularising effect that reduces overfitting to narrow typologies and improves generalisation across laundering techniques.

EXPERIMENTAL SETTINGS

This article proposes benchmarking Tab-AML against the TabTransformer and TabNet deep-learning models and five widely used classical baselines: logistic regression, k-nearest neighbours, decision tree (CART), random forest, and XGBoost. An 80-10-10 time-based train-validation-test split is proposed, with transactions chronologically ordered before splitting, so that the training, validation, and test sets each reflect a realistic, non-overlapping time window. The validation set would be used exclusively for hyperparameter tuning and the test set for final evaluation, with both monitored for signs of overfitting. The model would not be retrained on a combined training-and-validation set after tuning, to avoid introducing bias.

EVALUATION METRICS

Because transaction-monitoring data of this kind is typically highly imbalanced, accuracy would not be an appropriate evaluation metric on its own. ROC-AUC is proposed instead, as it captures a model's ability to discriminate between suspicious and normal transactions across all decision thresholds, ranging from 0.5 (random guessing) to 1 (perfect discrimination). For the best-performing models, ROC curves and confusion matrices at a fixed threshold, along with training and validation convergence curves for ROC-AUC and loss, would additionally need to be examined to understand each model's learning dynamics.

PROPOSED HYPERPARAMETER SEARCH: TAB-AML AND TABTRANSFORMER

Both models would use Binary Cross-Entropy with Logits Loss for numerical stability. This article proposes testing learning rates from 1e−5 to 1e−2, an AdamW optimiser (given its established general advantage over Adam), model dimensionality of 16, 32, 64, and 128, a multi-head configuration of either 4 heads with 2 layers or 8 heads with 4 layers, batch sizes of 32, 64, 128, and 256, and a dropout rate of 0%, 10%, or 20% (residual dropout and MLP dropout for Tab-AML; attention dropout and feed-forward dropout for TabTransformer). The specific optimal values within these ranges would need to be determined empirically for the specific dataset used.

Table 4. Proposed Hyperparameter Search Space for the Tab-AML and TabTransformer Models

Parameter	Proposed Search 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

Parameter	Proposed Search Values
Multi-Head Attention Config.	4 heads / 2 layers, 8 heads / 4 layers
Residual / Attention Dropout	0%, 10%, 20%
MLP / Feed-Forward Dropout	0%, 10%, 20%

PROPOSED HYPERPARAMETER SEARCH: TABNET

TabNet would be implemented in PyTorch with cross-entropy loss and the Adam optimiser. This article proposes testing decision-step depth and attention-layer width at 16, 32, and 64, a decision-step count of 1, 3, or 5, feature scaling (γ) of 1.0, 1.5, and 2.0, sparse regularisation of 1e−6, 1e−4, and 1e−2, batch sizes of 2048, 4096, and 16384, virtual batch sizes of 512, 1024, and 2048 for ghost batch normalisation, momentum of 0.1, 0.25, and 0.4, and a learning rate of 1e−2, 2e−2, or 3e−2. As with Tab-AML and TabTransformer, the specific optimal values would need to be determined empirically.

Table 5. Proposed Hyperparameter Search Space for the TabNet Model

Parameter	Proposed Search Values
Learning Rate (lr)	1e−2, 2e−2, 3e−2
Depth of Decision Steps (Nd)	16, 32, 64
Width of Attention Layers (Na)	16, 32, 64
Number of Decision Steps (Nsteps)	1, 3, 5
Feature Scaling (γ)	1.0, 1.5, 2.0
Sparse Regularisation (λsparse)	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

BASELINE MODELS

Five classical baselines widely used in the AML literature were included for comparison: logistic regression (lbfgs solver, L2 strength C ∈ {0.01, 0.1, 1, 10, 100}); k-nearest neighbours (N ∈ {2, 3, 5, 7, 9}); decision tree / CART (max depth D ∈ {2, 4, 8, 16}, min samples split S ∈ {4, 8, 12}); random forest (max depth D ∈ {2, 4, 8, 12}, max features F ∈ {2, 4, 8, 12}); and XGBoost (max depth D ∈ {2, 4, 8, 16}, learning rate L ∈ {0.1, 0.2, 0.3}). Gaussian Naive Bayes is proposed for inclusion without tuning as an untuned benchmark. Each tuned baseline would use GridSearchCV with three-

fold cross-validation, scored on ROC-AUC.

RESULTS AND DISCUSSION

This section sets out the expected evaluation approach and the architectural reasoning behind it, since, as explained above, the specific numeric results any such evaluation would produce cannot be reported without first conducting it.

Under the proposed evaluation protocol described above, each model would be retrained after hyperparameter tuning and evaluated on the held-out test set using ROC-AUC as the primary metric, given the severe class imbalance the dataset would be expected to exhibit. Based on the architectural reasoning set out in Sections 4 and 5, and consistent with published evidence that transformer-based architectures have shown early promise for related, adjacent financial-crime detection tasks, Tab-AML is expected to outperform TabTransformer and TabNet on this evaluation, given its additional dual-masked encoder structure and residual-attention mechanism, which are specifically designed to capture the micro-level sender-receiver relationships central to layering and structuring typologies alongside macro-level transaction context. Among the classical baselines, XGBoost would be expected to be the strongest performer, consistent with its established reputation in the AML literature reviewed in Section 2, while logistic regression and k-nearest neighbours would be expected to trail behind the deep-learning approaches given their more limited capacity to model non-linear feature interactions. These expectations are grounded in architectural reasoning and the published literature reviewed throughout this article; confirming them empirically is the necessary next step this article's proposed evaluation protocol is designed to enable.

Tab-AML's expected advantage stems from its architectural design: the dual-masked encoder structure enables comprehensive analysis at both the micro level (sender-receiver interaction) and macro level (whole-feature context), while the shared-embedding mechanism improves the model's ability to learn feature interactions relevant to complex laundering typologies. The residual-attention mechanism further strengthens the model's ability to capture long-range dependencies across transactions, an important signal that traditional attention alone does not retain as effectively.

Expected Operational Trade-offs

For AML applications, a high true-positive rate is essential to avoid overlooking genuine suspicious activity, while minimising the false-positive rate is equally important to control investigative cost and avoid misallocating scarce compliance resources, particularly at community institutions with limited investigative headcount. The proposed evaluation protocol in Section 6 would examine ROC curves and confusion matrices at thresholds calibrated to a fixed, high true-positive rate for each model, since the practically relevant question for a resource-constrained institution is not overall discriminative power alone but how many false positives a model generates once it is tuned to catch a defined, high share of genuinely suspicious activity. If, as the architectural reasoning above suggests, Tab-AML's ROC curve dominates TabTransformer's across the threshold range rather than only at a single, cherry-picked operating point, this would indicate that any false-positive-rate improvement holds

broadly rather than being an artefact of one particular threshold choice, and would need to be confirmed empirically before being relied upon operationally.

IMPLICATIONS FOR BSA/AML PRACTICE

Deploying a model such as Tab-AML within a financial institution's transaction-monitoring programme, if the proposed evaluation confirms its architectural advantages empirically, would offer a concrete route to reduced alert volume without a corresponding reduction in detection coverage. Rather than replacing existing rule-based scenarios, which remain valuable for their transparency and auditability, Tab-AML is best positioned as an additional analytical layer applied to transactions or entities already alerted by rule-based logic, re-scoring and re-prioritising them so investigators can focus their limited time on the highest-risk cases. For a community financial institution, where compliance staffing is typically the binding constraint, even a modest reduction in false positives at a matched true-positive rate could materially shorten alert-review queues, freeing investigator time to conduct deeper research and produce more complete, better-documented SAR narratives on the cases that do warrant filing.

Deployment is not without challenges. Institutions adopting a model of this kind require integration with existing case-management and alerting infrastructure, access to sufficiently complete historical transaction data for training, and a compliance and technology team capable of validating and monitoring model performance on an ongoing basis, including periodic revalidation against regulatory model-risk-management expectations. Because Tab-AML is designed to operate on an institution's own transactional data within its existing secure infrastructure, it does not require sharing data externally, which helps mitigate data-privacy and confidentiality concerns relative to approaches that rely on pooled or third-party data.

CONCLUSION AND FUTURE WORK

This article proposed Tab-AML, a transformer-based decision-support model for BSA/AML transaction monitoring, designed to reduce the false-positive alert burden that disproportionately affects resource-constrained community financial institutions, while maintaining, and ideally improving on, the detection performance of existing rule-based and classical machine-learning approaches. Tab-AML combines a dual-masked transformer encoder, a residual-attention mechanism, and a shared-embedding scheme to model both micro-level sender-receiver relationships and macro-level transaction context. This article set out a detailed, proposed evaluation methodology using a large-scale synthetic transaction-monitoring benchmark, and outlined the specific hyperparameter search space, baseline comparisons against TabTransformer, TabNet, and five classical machine-learning models, and false-positive-rate evaluation protocol such a study would need to follow. If empirically confirmed, findings of this

kind would indicate that adaptive, transformer-based decision support can meaningfully lighten alert-queue burden without weakening suspicious-activity detection, directly supporting more thorough investigation and higher-quality SAR documentation. Because the model is designed to complement, rather than replace, existing rule-based controls, it offers a practical and incremental adoption path for institutions of varying size and technical maturity, including smaller community institutions that cannot undertake a full transaction-monitoring system replacement. Several directions remain for future work. Beyond the proposed synthetic-benchmark evaluation outlined here, validating Tab-AML on real, institution-specific transaction data would strengthen confidence in its real-world applicability. Further work could also explore alternative embedding schemes tailored to individual feature cardinalities, automated hyperparameter search using methods such as Optuna in place of the manual tuning strategy proposed here, and techniques to improve model explainability, for example surfacing feature-importance scores to investigators to help justify SAR decisions and satisfy examiner documentation expectations. Combining Tab-AML's outputs with large language models to help draft or review SAR narratives is another promising avenue for reducing investigator workload while maintaining, or improving, report quality.

REFERENCES

[1] Zhang, Y., & Trubey, P. (2019). Machine learning and sampling scheme: An empirical study of money laundering detection. *Computational Economics, 54*, 1043–1063.

[2] Lv, L. T., Ji, N., & Zhang, J. L. (2008). A RBF neural network model for anti-money laundering. In Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition (Vol. 1, pp. 209–215).

[3] Tundis, A., Nemalikanti, S., & Mühlhäuser, M. (2021). Fighting organized crime by automatically detecting money laundering-related financial transactions. In Proceedings of the 16th International Conference on Availability, *Reliability and Security (Vol. 38*, pp. 1–10).

[4] Ketenci, U. G., et al. (2021). A time-frequency based suspicious activity detection for anti-money laundering. *IEEE Access, 9*, 59957–59967.

[5] Shokry, A. E. M., Rizka, M. A., & Labib, N. M. (2020). Counter terrorism finance by detecting money laundering hidden networks using unsupervised machine learning algorithm. In Proceedings of the International Conference on e-Learning (pp. 89–97).

[6] Guevara, J., Garcia-Bedoya, O., & Granados, O. (2020). Machine learning methodologies against money laundering in non-banking correspondents. In Applied Informatics (pp. 72–88). *Springer International Publishing*.

[7] Jullum, M., et al. (2020). Detecting money laundering transactions with machine learning. *Journal of Money Laundering Control, 23*(1), 173–186.

[8] Stojanović, B., et al. (2021). Follow the trail: Machine learning for fraud detection in Fintech applications. *Sensors, 21*(5).

[9] Zhiyuan, C., et al. (2021). Variational autoencoders and Wasserstein generative adversarial networks for improving the anti-money laundering process. *IEEE Access, 9*, 83762–83785.

[10] Alshantti, A., & Rasheed, A. (2021). Self-organising map based framework for investigating accounts suspected of money laundering. *Frontiers in Artificial Intelligence, 4*, 1–15.

[11] Paula, E. L., et al. (2016). Deep learning anomaly detection as support fraud investigation in Brazilian exports and anti-money laundering. In Proceedings of the 15th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 954–960).

[12] Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020). Competence of graph convolutional networks for anti-money laundering in Bitcoin blockchain. In Proceedings of the 2020 5th International Conference on Machine Learning Technologies (pp. 23–27). Association for Computing Machinery.